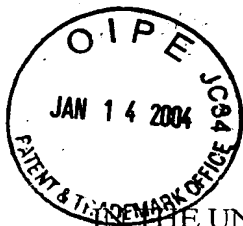


46056



PATENT

THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of :
Sang Hyuck Ha et al. :
Serial No.: 10/718,816 : Group Art Unit:
Filed: November 24, 2003 :
For: METHOD AND APPARATUS FOR :
CONTROLLING TURBO DECODER :
INPUT :

TRANSMITTAL OF PRIORITY DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In order to perfect the claim for priority under 35 U.S.C. §119(a), the Applicants herewith submit a certified copy of Korean Patent Application No. 2002-73280, as filed on November 23, 2002. Should anything further be required, the Office is asked to contact the undersigned attorney at the local telephone number listed below.

Respectfully submitted,

Peter L. Kendall
Attorney of Record
Reg. No.: 46,246

Roylance, Abrams, Berdo & Goodman, L.L.P.
1300 19th Street, N.W., Suite 600
Washington, D.C. 20036-2680
(202) 659-9076

Dated: January 14, 2004



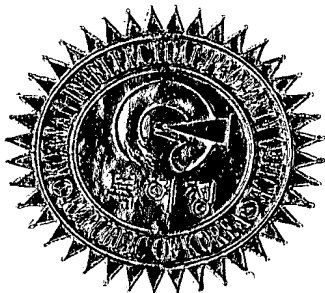
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0073280
Application Number

출원년월일 : 2002년 11월 23일
Date of Application NOV 23, 2002

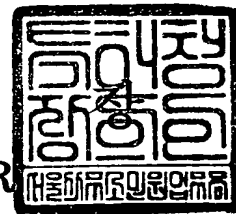
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 11 월 25 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2002. 11. 23
【국제특허분류】	G06F
【국제특허분류】	H04L
【발명의 명칭】	터보 복호기의 입력 제어 방법 및 장치
【발명의 영문명칭】	INPUT CONTROL METHOD AND APPARATUS FOR TURBO DECODER
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	하상혁
【성명의 영문표기】	HA, SANG HYUCK
【주민등록번호】	730219-1167429
【우편번호】	441-390
【주소】	경기도 수원시 권선구 권선동 1314번지 주공1단지아파트 121동 1003 호
【국적】	KR
【발명자】	
【성명의 국문표기】	유남열
【성명의 영문표기】	YU, Nam Yul
【주민등록번호】	711207-1024118
【우편번호】	442-725
【주소】	경기도 수원시 팔달구 영통동 벽적골8단지 아파트 812-804
【국적】	KR

【발명자】

【성명의 국문표기】

김민구

【성명의 영문표기】

KIM,MIN G00

【주민등록번호】

640820-1067025

【우편번호】

442-811

【주소】

경기도 수원시 팔달구 영통동 968 신나무실 신명아파트
633-1502

【국적】

KR

【취지】

특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인
이건주 (인)

【수수료】

【기본출원료】

20 면 29,000 원

【가산출원료】

50 면 50,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

0 항 0 원

【합계】

79,000 원

【요약서】**【요약】**

본 발명은 1xEV-DV 표준을 지원하는 단말기에서 순방향 패킷 데이터 트래픽의 복호를 위해 터보 복호기로의 입력 데이터를 효과적으로 제어하기 위한 방법 및 장치에 관한 것이다. 순방향 패킷 데이터 트래픽의 복호는 정해진 개수의 부호 심볼들을 적절한 순서로 터보 복호기에 입력함으로써 시작된다. 본 발명은 순방향 패킷 데이터 트래픽의 복조 결과를 복호기 입력 버퍼로부터 적절한 순서로 읽어낸 뒤 터보 복호기 입력을 위한 적합한 형식의 데이터를 생성하는 복호기 입력 제어 장치를 제공한다. 이러한 본 발명은 채널 인터리빙된 데이터를 고속으로 처리하여 복호기로 입력함으로써 필요한 버퍼의 개수를 줄여 소형화, 저가격화, 저전력화를 실현하며, 지연을 감소시켜 고속의 데이터 서비스를 지원할 수 있도록 한다.

【대표도】

도 10

【색인어】

1xEV-DV, packet data, turbo decoding, channel interleaving, encoder packet

【명세서】**【발명의 명칭】**

터보 복호기의 입력 제어 방법 및 장치{INPUT CONTROL METHOD AND APPARATUS FOR TURBO DECODER}

【도면의 간단한 설명】

도 1은 순방향 패킷 데이터 채널을 위한 패킷 데이터를 인터리빙하여 송신 서브패킷을 생성하는 기지국 송신기의 구성도.

도 2는 순방향 패킷 데이터 채널을 통해 수신된 서브패킷으로부터 복호된 데이터를 얻는 단말 수신기의 구성도.

도 3은 상기 도 1에 나타낸 채널 인터리빙 과정을 상세히 도시하는 도면.

도 4는 모부호율 1/5인 터보 부호기의 구성과 출력 부호심볼들의 예를 나타낸 도면.

도 5는 상기 도 4에서 나타낸 터보 부호기의 출력 심볼들이 채널 인터리빙 되는 과정을 나타낸 도면.

도 6은 모부호율 1/5인 터보 복호기의 구성과 입력 부호심볼들의 예를 나타낸 도면.

도 7은 1xEV-DV 표준의 순방향 패킷 데이터 트래픽 채널을 위해 사용 가능한 부호화 패킷의 크기에 대해 사용 가능한 부호심볼들을 도시한 도면.

도 8은 1xEV-DV 표준의 순방향 패킷 데이터 트래픽을 위한 복호기 입력 버퍼의 구조를 도시하는 도면.

도 9는 본 발명의 일 실시예에 따른 터보 복호기를 위한 입력 제어 장치의 구성을 개략적으로 도시한 블록도.

도 10은 상기 도 9에 나타난 터보 복호기를 위한 입력 제어 장치의 동작을 나타낸 흐름도.

도 11은 본 발명에 따른 터보 복호기를 위한 입력 제어 장치에서 읽기 주소 생성기의 동작을 나타낸 흐름도.

도 12는 본 발명에 따른 터보 복호기를 위한 입력 제어 장치에서 읽기 주소 생성기의 상세 구성을 나타낸 블록도.

도 13은 본 발명에 따른 읽기 주소 생성기에서 주소 조절기의 동작을 나타낸 흐름도.

도 14는 본 발명에 따른 읽기 주소 생성기에서 주소 조절기의 상세 구성을 나타낸 블록도.

도 15는 본 발명에 따른 읽기 주소 생성기에서 메모리 선택기의 동작을 나타낸 흐름도.

도 16은 메모리 선택기에서 N_EP이 3인 경우 RAM1_CS와 RAM2_CS를 결정하는 동작을 나타낸 흐름도.

도 17은 메모리 선택기에서 N_EP이 3보다 큰 경우 RAM1_CS와 RAM2_CS를 결정하는 동작을 나타낸 흐름도.

도 18은 본 발명에 따른 읽기 주소 생성기에서 메모리 선택기의 상세 구성을 나타낸 블록도.

도 19는 본 발명에 따른 터보 복호기를 위한 입력 제어 장치에서 메모리부의 동작을 나타낸 흐름도.

도 20은 본 발명에 따른 터보 복호기를 위한 입력 제어 장치에서 메모리부의 상세 구성을 나타낸 블록도.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <21> 본 발명은 고속 패킷 데이터 통신 시스템에 관한 것으로서, 특히 수신된 데이터를 터보 복호기로 입력하는 방법 및 장치에 관한 것이다.
- <22> CDMA(Code Division Multiple Access)2000 및 UMTS(Universal Mobile Telecommunication Service)와 같은 광대역(Wide-band) CDMA 방식의 이동통신 시스템에서는 멀티미디어 데이터의 신뢰성 있는 고속 전송을 위해 터보 부호화(Turbo Coding)와 인터리빙(Interleaving)을 사용한다. 터보 부호화는 낮은 신호대 잡음비에서도 비트 에러율(Bit Error Rate: BER) 관점에서 매우 양호한 정보 복원 성능을 보이는 것으로 알려져 있다. 인터리빙은 페이딩 환경에서 인접한 비트들이 랜덤 하게 페이딩 영향을 받도록 함으로서 군집에러(burst error)가 발생하지 않도록 해 주어 채널 부호화의 효과를 보다 높여준다.
- <23> 3GPP(3rd Generation Partnership Project) 및 3GPP2에서 추진중인 CDMA2000 Release C, 이른 바 1xEV-DV(Evolution in Data and Voice) 표준안에 따르면, 기지국은 전송하고자 하는 패킷 데이터의 열을 채널 부호기(Channel Encoder)에 의해 부호화하여 생성한 부호심볼들(Code Symbols)을 종류에 따라 소정 크기의 서브블럭(subblock)들로 분할하고, 상기 분리된 서브블럭들을 각각 인터리빙한다. 단말기는 상기 인터리빙된 데이터의 열을 수신하여 부호심볼의 형태

로 변환한 후 기지국에서 사용한 인터리빙의 역과정인 서브블럭 디인터리빙(Subblock deinterleaving)을 거쳐 채널 부호기의 출력과 동일하게 복원하며, 상기 복원된 데이터를 터보 복호기(Turbo Decoder)에 의하여 복호한다.

<24> 1xEV-DV에서는 기지국의 터보 부호기에서 한번에 부호화되는 부호화 패킷(Encoder Packet)의 크기를 408,792,1560,2328,3096,3864비트의 6가지 중 하나로 규정하고 있으며, 순방향 패킷 데이터 채널(Forward Packet Data Channel: F-PDCH)의 한 시간구간동안 전송 가능한 부호심볼들의 최대 개수를 7800개로 제한하고 있다. 게다가 터보 복호시의 성능을 최대화하기 위하여 기지국에서 전송되는 부호심볼들은 다른 종류의 부호심볼들과 엇갈리도록 배치되어 전송된다. 이와 같이 복잡한 절차를 거쳐 전송되는 데이터를 수신측에서 정확하게 복원하기 위해서는 각각의 처리를 위한 별도의 버퍼들이 필요하며 이에 따른 지연이 발생한다.

<25> 단말 수신기에서 사용되는 이들 버퍼들은 주로 랜덤 액세스 메모리(Random Access Memory: 이하 "RAM"이라 한다.)로 구성되어 있으며 각 버퍼의 입/출력은 순차적으로 이루어지기 때문에, 버퍼들의 개수가 증가할수록 전체 데이터 처리시간이 큰 폭으로 증가한다. 그러므로, 효율적인 버퍼관리는 고속 데이터 서비스를 지원하는 이동국 수신기를 설계하는데 있어서 중요한 고려사항이 된다. 따라서 특히 고속의 데이터 서비스를 제공하는 이동통신시스템에서 터보 복호기로 입력되는 데이터를 정확하고 신속하게 복원하기 위한 기술을 필요로 하게 되었다.

【발명이 이루고자 하는 기술적 과제】

- <26> 따라서 본 발명의 목적은 1xEV-DV 표준을 지원하는 통신시스템의 수신단에서 순방향 패킷 데이터 트래픽에 대한 복호에 앞서 채널 디인터리빙을 수행하는 방법 및 장치를 제공함에 있다.
- <27> 본 발명의 다른 목적은 1xEV-DV 표준을 지원하는 통신시스템의 수신단에서 터보 복호기의 입력버퍼를 이용하여 채널 디인터리빙을 수행하는 방법 및 장치를 제공하는 것이다.
- <28> 본 발명의 다른 목적은 1xEV-DV 표준을 지원하는 단말기에서 순방향 패킷 데이터 트래픽에 대한 터보 복호기의 입력 데이터를 구성하는 방법 및 장치를 제공함에 있다.
- <29> 본 발명의 다른 목적은 1xEV-DV 표준을 지원하는 통신시스템의 수신단에서 수신된 부호심볼들을 저장하고 순서대로 독출하여 복호기로 입력하는 방법 및 장치를 제공하는 것이다.
- <30> 상기 목적들을 달성하기 위한 본 발명의 일 실시예에 따른 방법은, 송신기에서 소정 개수의 데이터 비트들과 테일 비트들을 포함하는 부호화 패킷을 소정 부호율에 따라 부호화하여 생성한 5개의 서브블럭들의 부호심볼들을 인터리빙하여 전송하면, 수신기에서 상기 부호심볼들을 수신하여 저장하고 상기 인터리빙에 대응하는 디인터리빙 순서대로 읽어내어 터보 복호기로 입력하는 방법에 있어서,
- <31> 상기 수신된 부호심볼들은 종류에 따라 3개의 메모리들 RAM0, RAM1, RAM2에 나누어 저장되고, 상기 터보 복호기는 순차적으로 동작하면서 상기 3개의 메모리들로부터 읽어낸 3개의 부호심볼들을 복호하는 2개의 구성 복호기들로 구성되며,

- <32> 상기 터보 복호기로부터 수신되는 부호화 패킷의 크기와 복호기 클럭과 복호기 인덱스 DEC_IDX를 이용하여 상기 3개의 메모리들을 위한 읽기주소들 RAMx_ADDR과 칩 선택 신호들 RAMx_CS를 생성하는 과정과,
- <33> 상기 생성된 읽기주소들 RAMx_ADDR과 상기 칩 선택 신호들 RAMx_CS를 이용하여 상기 2개의 구성 복호기들 중 하나로 입력될 3개의 부호심볼들을 출력하는 과정을 포함한다.
- <34> 본 발명의 일 실시예에 따른 장치는, 송신기에서 소정 개수의 데이터 비트들과 테일 비트들을 포함하는 부호화 패킷을 소정 부호율에 따라 부호화하여 생성한 5개의 서브블럭들의 부호심볼들을 인터리빙하여 전송하면, 수신기에서 상기 부호심볼들을 수신하여 저장하고 상기 인터리빙에 대응하는 디인터리빙 순서대로 읽어내어 터보 복호기로 입력하는 장치에 있어서,
- <35> 상기 수신된 부호심볼들은 종류에 따라 3개의 메모리들 RAM0,RAM1,RAM2에 나누어 저장되고, 상기 터보 복호기는 순차적으로 동작하면서 상기 3개의 메모리들로부터 읽어낸 3개의 부호심볼들을 복호하는 2개의 구성 복호기들로 구성되며,
- <36> 상기 터보 복호기로부터 수신되는 부호화 패킷의 크기와 복호기 클럭과 복호기 인덱스 DEC_IDX를 이용하여 상기 3개의 메모리들을 위한 읽기주소들 RAMx_ADDR과 칩 선택 신호들 RAMx_CS를 생성하는 읽기주소 생성기와,
- <37> 상기 생성된 읽기주소들 RAMx_ADDR과 상기 칩 선택 신호들 RAMx_CS를 이용하여 상기 2개의 구성 복호기들 중 하나로 입력될 3개의 부호심볼들을 출력하는 메모리부를 포함한다.

【발명의 구성 및 작용】

- <38> 이하 본 발명을 설명함에 있어서, 관련된 공지 기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략한다. 또한, 동일한 부분은 비록 다른 도면에 도시되더라도 동일한 참조부호를 사용한다.
- <39> 이하 본 발명은 1xEV-DV 표준을 지원하는 단말기에서 순방향 패킷 데이터 트래픽의 복호를 수행하는 터보 복호기를 위한 입력을 제어하는 것이다.
- <40> 먼저, CDMA2000 1xEV-DV 이동통신 시스템에서 패킷 데이터의 열을 송신하고, 수신된 데이터의 열을 복호하기 위한 구성 및 그 동작을 설명하기로 한다.
- <41> 도 1은 순방향 패킷 데이터 채널을 위한 패킷 데이터를 인터리빙하여 송신 서브패킷(Transmission Subpacket)을 생성하는 기지국 송신기의 구성을 나타낸 것이다.
- <42> 상기 도 1을 참고하면, 터보 부호기(Turbo Encoder)(10)는 입력되는 패킷 데이터의 열을 부호화 패킷(Encoder Packet) 단위로 소정 부호율(Code Rate) R에 따라 부호화하여 부호심볼들(Code Symbols)의 열을 출력한다. 여기서 상기 부호심볼들은 그 종류에 따라, 전송하고자 하는 데이터인 시스템틱 심볼(Systematic Symbol)과 상기 데이터의 복원을 위한 에러 정정 정보인 패리티 심볼(Parity Symbol)로 구분된다.
- <43> 1/5의 부호율이 사용되는 경우 상기 터보 부호기(10)는 매 입력 비트에 대하여 5개의 부호심볼들을 출력하는 한편, 복호시에 특정 상태로의 수렴을 위해 가상으로 생성된 소정 개수의 테일비트들(tail bits)에 대한 부호심볼들을 더 출력하게 된다. 여기서 가상으로 생성되었다는 의미는 상기 테일비트들이 실제로 입력되지는 않지만 그에 대한 부호심볼들이 출력됨을 뜻한다. 1xEV-DV 표준에서 규정하고 있는 부호화 패킷의 크기(EP_SIZE)는 408, 792, 1560, 2328,

3096, 3864의 6가지이며, 하나의 부호화 패킷을 위하여 6개의 테일 비트들을 생성하도록 규정하고 있다. 따라서 상기 규정된 부호화 패킷의 크기들에 대해 실제로 부호화되는 데이터의 크기는 각각 402, 786, 1554, 2322, 3090, 3858비트이다.

<44> 채널 인터리버(Channel Interleaver)(18)는 서브패킷을 형성하는 부호심볼들이 높은 수준의 부호화이득(coding gain)을 보장할 수 있도록, 상기 터보 부호기(10)로부터 출력되는 부호심볼들을 적절히 배열한다. 서브패킷 심볼 선택기(Subpacket Symbol Selector)(20)는 상기 채널 인터리버(18)로부터의 부호심볼들 전체 또는 일부를 선택하여 소정 크기의 서브패킷을 구성한다.

<45> 상기 채널 인터리버(18)의 동작은 심볼 분리(Symbol Separation)(12), 서브블럭 인터리빙(Subblock interleaving)(14), 서브블럭 심볼 그룹화(Subblock symbol grouping)(16)를 차례로 수행함으로써 이루어진다. 보다 상세히 설명하면, 심볼 분리기(12)는 상기 터보 부호기(10)로부터 출력되는 부호심볼들을 종류대로 분류하여 서브블럭들을 구성하고, 서브블럭 인터리버(14)는 상기 구성된 서브블럭들 각각을 동일한 인터리빙 규칙(interleaving rule)에 따라 인터리빙하며, 서브블럭 심볼 그룹화기(16)는 상기 인터리빙된 서브블럭들의 부호심볼들을 서로 엇갈리도록(alternately) 배치하여 그룹화한다. 상기 채널 인터리버(18)의 동작에 대한 보다 상세한 설명은 후술될 것이다.

<46> 순방향 패킷 데이터 채널을 통해 기지국이 송신하는 고속의 패킷 데이터를 수신하는 단말은 상기 도 1에 나타난 동작을 역으로 수행함으로써 복호된 데이터를 얻는다.

<47> 도 2는 순방향 패킷 데이터 채널을 통해 수신된 서브패킷으로부터 복호된 데이터를 얻는 단말 수신기의 구성을 나타낸 것이다.

- <48> 상기 도 2를 참조하면, 서브패킷 제로 삽입기(Subpacket Zero Inserter)(22)는 기지국 송신기의 심볼 선택기(20)에 대응하여, 수신된 서브패킷의 정해진 위치에 부호심볼 '0'을 삽입한다. 기지국 송신기의 채널 인터리버(18)에 대응되는 단말기 수신기의 채널 디인터리버(channel deinterleaver)(30)는 서브블럭 심볼 역그룹화(Subblock Symbol Ungrouping)(24), 서브블럭 디인터리빙(Subblock Deinterleaving)(26), 심볼 역분리(Symbol Deseperation)(28)를 차례로 수행한다.
- <49> 보다 상세히 설명하면, 서브블럭 심볼 역그룹화기(24)는 상기 서브패킷 제로 삽입기(22)로부터의 출력을 종류별로 분리하여 5개의 인터리빙된 서브블럭들을 출력한다. 서브블럭 디인터리버(26)는 상기 인터리빙된 서브블럭들 각각을 기지국 송신기의 서브블럭 인터리버(14)에서 사용된 인터리빙 규칙에 대응하는 디인터리빙 규칙에 따라 디인터리빙하고, 심볼 역분리기(28)는 상기 디인터리빙된 출력을 모아 터보 복호기(Turbo Decoder)(32)로 제공한다. 상기 터보 복호기(32)는 기지국 송신기의 터보 부호기(10)에서와 동일한 부호율 R을 가지고 상기 디인터리빙된 출력을 복호하여 복호된 데이터를 생성한다.
- <50> 도 3에 상기 채널 인터리버(18)에 의한 채널 인터리빙 동작을 상세하게 도시하였다.
- <51> 도시된 바와 같이 상기 터보 부호기(10)로부터의 부호심볼들은 심볼 분리기(12)에 의해 종류에 따라 5개의 서브블럭들 S, P0, P'0, P1, P'1(40 내지 48)로 분리된다. 여기서 S는 시스템릭 서브블럭이며, P0, P'0, P1, P'1은 패리티 서브블럭들이다. 상기 서브블럭들 각각은 서브블럭 인터리버(14)에 의해 소정 인터리빙 규칙에 따라 인터리빙된다. (50 내지 58) 서브블럭 심볼 그룹화기(16)에서는 상기 인터리빙된 서브블럭들(50 내지 58) 중 인터리빙된 패리티 서브블럭들(52 내지 58)의 심볼들을 서로 엇갈리도록 배치하여 그룹화한다. 즉 인터리빙된 서브블럭 P0(52)과 인터리빙된 서브블럭 P'0(54)의 부호심볼들을 서로 엇갈리도록 배치하여 제1 데이터

그룹(52)을 생성하고, 인터리빙된 서브블럭 P1(56)과 인터리빙된 서브블럭 P'1(58)의 부호심볼들을 서로 엇갈리도록 배치하여 제2 데이터 그룹(54)을 생성한다.

<52> 상기에 나타낸 바와 같이 채널 부호기(10)로부터의 부호심볼들은 서브 블록 분리, 서브블럭 인터리빙, 서브블럭 심볼 그룹화 등의 복잡한 절차를 거쳐 인터리빙된다. 수신단에서 이러한 부호심볼들을 복호하기 위해 다시 서브블럭 심볼 역그룹화, 서브블럭 디인터리빙, 심볼 역분리 등을 각각 수행하게 되면, 각 동작별로 버퍼들이 필요하게 되어 처리 시간이 매우 증가하게 된다. 따라서 본 발명에서는, 수신단에서 매 서브패킷을 수신할 때마다 상기 서브패킷의 부호심볼들을 버퍼에 저장하고, 이들을 터보 부호기(10)에서 출력한 원래의 순서대로 독출하여 복호기로 입력함으로써, 서브블럭 심볼 역그룹화와 서브블럭 디인터리빙과 심볼 역분리를 동시에 수행한다.

<53> 먼저 본 발명에 대한 이해를 돕기 위해 먼저 송신 서브패킷이 어떠한 형태로 구성되는지 간략화된 예를 들어 설명하기로 한다.

<54> 도 4는 1/5의 모부호율(mother code rate)을 가지는 터보 부호기(10)의 상세 구성과 상기 터보 부호기(10)로부터 출력되는 부호심볼들을 나열한 것이다. 여기서 모부호율이란 상기 서브패킷 심볼 선택기(20)를 통과하기 전의 공통적인 부호율을 의미하는 값으로 1xEV-DV 표준에서 그 값을 1/5로 정하고 있다. 또, 여기서 입력 데이터의 비트 수는 6이며 가상으로 생성되는 테일비트의 개수는 총 4개인 것으로 한다. 즉, 상기 터보 부호기(10)에서 처리되는 부호화 패킷의 비트 수는 10개이다.

- <55> 상기 도 4를 참조하면, 상기 터보 부호기(10)는 하나의 터보 인터리버(Turbo Interleaver) TB INTL(10a)와 2개의 구성 부호기들(Constituent Encoders) (Encoder 1: ENC1)(10b), (Encoder 2: ENC2)(10c)로 구성된다. 상기 구성 부호기들(10b,10c)은 1/3의 부호율을 가지며 부호화기 클럭 t에 동기하여 입력 데이터 비트들 $X_0 \sim X_5$ 각각에 대해 1개의 시스템틱 부호심볼 SYS와 2개의 패리티 부호심볼들 PA0, PA1을 출력하고, 데이터 비트들의 부호화가 완료되면 각각 2개씩의 테일비트들을 부호화한다.
- <56> 상기 도 4에서 X는 출력된 후 바로 제거되는 부호심볼들을 나타내며 ??는 출력되지 않는 부호심볼들을 나타내는 것으로, 데이터 비트들이 입력되는 t=0부터 t=5까지는 ENC2(10c)로부터의 SYS 심볼들이 출력된 후 바로 제거되고 t=6, t=7에서는 ENC2(10c)로부터의 모든 부호심볼들이 출력되지 않으며 t=8, t=9에서는 반대로 ENC1(10b)로부터의 모든 부호심볼들이 출력되지 않는다.
- <57> 결국 상기 터보 부호기(10)로부터 실제로 생성되는 출력 부호심볼들의 개수는 6개의 데이터 비트 입력에 대한 부호심볼들 30(=6*5)개와 4개의 테일비트 입력에 대한 부호심볼들 12(=4*3)개로 모두 36개이다. 그러나, 터보 부호기의 모부호율 1/5를 만족하기 위해서는 10개의 입력 비트들(데이터 비트 6개 + 테일 비트 4개)에 대해 50개의 부호심볼들이 출력되어야 하므로, t=6부터 t=9까지에서는 매 클럭마다 SYS 심볼들을 각각 두 번씩 복사하여 세 개씩의 SYS 심볼들을 출력한다. 즉, t=6, t=7에서는 ENC1(10b)의 SYS 심볼들 \hat{x}_0, \hat{x}_1 을 두 번 복사하고, t=8, t=9에서는 ENC2(10c)의 SYS 심볼들 \hat{x}'_0, \hat{x}'_1 을 각각 두 번씩 복사한다.
- <58> 도 5는 상기 도 4에 나타난 터보 부호화기(10)로부터 출력되는 부호심볼들의 인터리빙 동작을 상세히 나타낸 것이다.

<59> 도시한 바와 같이, 심볼 분리에 의해서, 상기 터보 부호기(10)의 출력 부호심볼들은 그 종류에 따라 다섯 개 서브블럭들 S, P0, P1, P'0, P'1로 분리된다. 여기서 서브블럭 S는 데이터 비트들에 대한 ENC1(10b)로부터의 SYS 부호심볼들과 테일 비트들에 대한 ENC1(10b)로부터의 SYS 부호심볼들과 테일 비트들에 대한 ENC2(10c)로부터의 SYS 부호심볼들을 포함한다. 서브블럭 P0는 데이터 비트들에 대한 ENC1(10b)의 PA0 부호심볼들과 테일 비트들에 대한 ENC1(10b)의 SYS 부호심볼들과 테일 비트들에 대한 ENC2(10c)의 SYS 부호심볼들을 포함한다. 서브블럭 P1은 데이터 비트들에 대한 ENC1(10b)로부터의 PA1 부호심볼들과 테일 비트들에 대한 ENC1(10b)로부터의 SYS 부호심볼들과 테일 비트들에 대한 ENC2(10c)의 SYS 부호심볼들을 포함한다. 서브블럭 P'0은 데이터 비트들에 대한 ENC2(10c)로부터의 PA0 부호심볼들과 테일 비트들에 대한 ENC1(10b)로부터의 PA0 부호심볼들과 테일 비트들에 대한 ENC2(10c)의 PA0 부호심볼들을 포함한다. 서브블럭 P'1은 데이터 비트들에 대한 ENC2(10c)로부터의 PA1 부호심볼들과 테일 비트들에 대한 ENC1(10b)의 PA1 부호심볼들과 테일 비트들에 대한 ENC2(10c)의 PA1 부호심볼들을 포함한다.

<60> 상술한 바와 같이 테일 비트들에 대한 SYS 부호심볼들 때문에 각 서브블럭은 다른 종류의 부호심볼들을 일부 포함하게 된다. 즉, 서브블럭 S은 ENC2(10c)로부터의 SYS 심볼을, 서브블럭 P0은 ENC1(10b)로부터의 SYS 심볼들과 ENC2(10c)로부터의 SYS 심볼들을, 서브블럭 P1은 ENC1(10b)로부터의 SYS 심볼들과 ENC2(10c)로부터의 SYS 심볼들을 더 포함하고 있다. 또한, 서브블럭 P'0은 ENC1(10b)의 PA0 심볼들을, 서브블럭 P'1은 ENC1(10b)로부터의 PA1 심볼들을 더 포함하고 있다.

- <61> 상기 분리된 서브블럭들은 서브블럭 인터리빙에 의해 재배열된다. 도시한 바에 따르면 0, 1, 2, 3, 4, 5, 6, 7, 8, 9의 순서로 배열되어 있던 각 서브블럭의 부호심볼들은 0, 8, 4, 2, 6, 1, 9, 5, 3, 7의 순서로 재배열되었다.
- <62> 서브블럭 심볼 그룹화에 의해서, 서브블럭 P0과 서브블럭 P'0의 부호심볼들을 하나씩 차례로 교차시켜 P0&P'0 데이터 그룹을 만들고, 서브블럭 P1과 서브블럭 P'1의 부호심볼들을 하나씩 차례로 교차시켜 P1&P'1 그룹을 만든다. 상기 생성된 서브블럭 S, P0&P'0 그룹, P1&P'1 그룹의 부호심볼들은 순서대로 변조기에 공급된다.
- <63> 도 6은 1/5의 모부호율을 가지는 터보 복호기(32)의 구조와 상기 터보 복호기(32)로 입력되는 부호심볼들을 나열한 것이다. 실제로 터보 복호는 복호 성능을 보다 향상시키기 위하여, 복호된 데이터를 피드백하여 다시 복호하는 반복 복호(iterative decoding)을 포함하나 상기 도 6에서는 설명의 편의를 위하여 이를 생략하고 있다.
- <64> 상기 도 6을 참조하면, 상기 터보 복호기(32)는 1/3의 부호율을 가지며 복호기 클럭 t_1 , t_2 에 각각 동기하는 두 개의 구성 복호기(Constituent Decoder) DEC1(32a)과 DEC2(32d)를 순차적으로 구동시킴으로써 입력 부호심볼들을 복호한다. DEC1(32a)는 $t_1=0$ 부터 $t_1=7$ 까지에서 ENC1(10b)로부터의 출력 부호심볼들을 한번에 3개씩 8번에 걸쳐 입력받는다.
- <65> DEC2(32d)는 $t_2=0$ 부터 $t_2=7$ 까지에서 ENC2(10c)로부터의 출력 부호심볼들과 0을 포함한 24개의 부호심볼들을 한번에 3개씩 8번에 걸쳐 입력받는다. 실제로 $t_2=0$ 부터 $t_2=5$ 까지에서 상기 DEC2(32d)는 터보 인터리버 TB INTL(32b)에 의해 인터리빙된 상기 DEC1(32a)의 복호된 데이터를 입력받게 되고, $t_2=6$, $t_2=7$ 에서만 ENC2(10c)로부터의 출력 부호심볼들 \hat{x}'_0 , \hat{x}'_1 을 입력받는다. 터보 디인터리버 TB DINTL(Turbo interleaver-32e)는 상기 DEC2(32d)에 의해 복호된 데이터를 디인터리빙하여 최종 복호된 데이터를 출력한다.

- <66> 상기에서 상세히 설명한 바와 같이, 인터리빙된 서브블럭들이 포함하고 있는 데이터 비트의 부호심볼과 테일비트의 부호심볼은 서로 다른 종류일 수 있으며, 터보 부호기(10)는 터보 복호기(32)의 총 구동 클럭보다 적은 수의 구동 클럭을 사용하는 2개의 구성 복호기를 순차적으로 구동시킨다. 즉, 터보 부호기(10)는 $t=0$ 부터 $t=9$ 까지 총 10회 동작하면서 부호심볼들을 생성하는 반면 터보 복호기(32)는 각 구성 복호기를 $t_1=0$ 부터 $t_1=7$ 까지 혹은 $t_2=0$ 부터 $t_2=7$ 까지 8회씩 순차적으로 동작시키면서 복호를 수행한다.
- <67> 결국, 1xEV-DV 표준을 지원하는 단말기에서 순방향 패킷 데이터 트래픽의 복호를 위해서는, 도 5에 나타난 순서대로 버퍼에 저장되어 있는 부호심볼들을 도 6에 나타난 순서대로 터보 복호기에 공급하여야 한다. 따라서 본 발명에서는 복호기 입력 버퍼(Decoder Input Buffer)에서 부호심볼들을 읽어낼 때 채널 디인터리빙 과정을 등가로 수행하여 터보 복호기로 바로 공급하기로 한다.
- <68> 한편, 1xEV-DV 표준에 따르면 서브패킷을 생성하기 위해서는 채널 인터리빙을 통해 출력되는 인터리빙된 부호심볼들 전체를 사용할 수 있는 것은 아니다. 도 1에 나타난 서브패킷 심볼 선택기(20)는 인터리빙된 부호심볼들 중 사용 가능한 일부만을 사용하여 서브패킷을 구성한다. 여기서 상기 사용 가능한 부호심볼들은 부호화 패킷의 크기(EP_SIZE)에 따라 도 7에 나타난 바와 같이 규정된다.
- <69> 상기 도 7에 나타난 바와 같이, 서브패킷을 위해 사용 가능한 부호심볼들의 개수는 부호화 패킷의 크기가 1560 이하인 경우(408,792,1560)에는 부호화 패킷 크기의 5배로 제한되며, 부호화 패킷의 크기가 1560을 초과하는 경우(2328,3096,3864)에는 부호화 패킷의 크기에 관계없이 7800개로 제한된다. 따라서, 순방향 패킷 데이터 트래픽을 수신하는 단말기가 필요로 하는 복호기 입력 버퍼는 최소한 7800개의 부호심볼들을 저장할 수 있는 크기를 가져야 한다.

- <70> 또한, 단말기에서 고속의 데이터를 처리하기 위해서는, 복호기로의 부호심볼 입력 시간을 최소화하기 위해 구성 복호기당 필요한 세 종류의 부호심볼들 SYS, PA0, PA1을 병렬로 전달한다. 이를 위해서 복호기 입력 버퍼는 세 개의 부호심볼들을 병렬로 출력할 수 있도록 세 개의 메모리들로 분리되고, 복조기에 의해 복원된 부호심볼들은 그 종류에 따라 각각 서로 다른 메모리에 저장된다.
- <71> 도 8은 1xEV-DV 표준의 순방향 패킷 데이터 트래픽을 위한 복호기 입력 버퍼의 구성을 나타낸 것이다.
- <72> 도시한 바와 같이, 입력 버퍼는 크기가 각각 3864, 3120, 3120인 RAM0, RAM1, RAM2로 이루어져 있다. 상기 RAM0은 서브블럭 S의 부호심볼들을 저장하고, 상기 RAM1은 P0&P'0 그룹이나 서브블럭 P0의 적어도 일부 혹은 서브블럭 P'1의 일부를 저장하며, 상기 RAM2는 P1&P'1 그룹이나 서브블럭 P'0의 적어도 일부 혹은 서브블럭 P1의 일부를 저장한다. 이와 같이 복호기 입력 버퍼의 구조가 다소 복잡한 이유는, 부호화 패킷의 크기에 따라 사용 가능한 부호심볼의 종류가 제한되어 있고, 복호기로의 부호심볼 입력 시간을 최소화하기 위해서이다.
- <73> 이하 상기 도 8에 나타낸 바와 같이 구성된 복호기 입력 버퍼로부터 부호심볼들을 순서대로 읽어내기 위한 방법 및 장치에 대하여 상세히 설명한다.
- <74> 도 9는 본 발명의 일 실시예에 따른 터보 복호기를 위한 입력 제어 장치의 구성을 개략적으로 도시한 블록도이다. 도시된 바와 같이 입력 제어 장치는, 읽기주소 생성기(Read Address Generator: RAG)(100)와, 입력 버퍼(210, 220, 230)를 포함하는 메모리부(Memory Part)(200)를 포함하여 구성된다. 상기 읽기주소 생성기(100)는 터보 복호기로부터 입력되는

신호들에 응답하여 상기 입력 버퍼를 위한 3개의 읽기주소들을 출력하며, 상기 메모리부(200)는 상기 읽기주소들에 응답하여 3개의 부호심볼들을 상기 터보 복호기로 출력한다.

<75> 상기 읽기주소 생성기(100)으로 입력되는 신호들에 대해 설명하면, DEC_START는 터보 복호기가 동작을 시작함을 알리는 신호이며, DEC_END는 터보 복호기가 하나의 서브패킷에 대한 복호를 완료하였음을 알리는 신호이고, DEC_CLK은 터보 복호기의 동작 주파수에 맞춰 공급되는 클럭이다. 또한, EP_SIZE는 터보 복호를 수행해야 할 부호화 패킷의 크기(N_EP)를 3개의 비트로 표시한 것으로 408, 792, 1560, 2328, 3096, 3864에 대해 각각 '000', '001', '010', '011', '100', '101'의 값을 가진다. DEC_IDX는 터보 복호기 내부에 위치한 두 개의 구성 복호기들 중 어떤 복호기가 동작하고 있는지를 표시하는 것으로 첫 번째 구성 복호기가 동작할 때는 0, 두 번째 구성 복호기가 동작할 때는 1의 값을 가진다.

<76> 도 10은 도 9에 나타낸 입력 제어 장치의 동작을 나타낸 흐름도이다.

<77> 상기 도 10에서 흐름도의 시작 전에 3개의 메모리들(210,220,230)에는 복조기에 의해 복조된 한 서브패킷의 부호심볼들이 인터리빙된 순서대로 저장되어 있어야 한다. 여기서 상기 3개의 메모리들(210,220,230)에 저장되는 부호심볼들의 종류는 앞서 언급한 도 8에 나타낸 바와 같다.

<78> 먼저, 과정(S100)에서 DEC_START가 '1'이면 과정(S200)에서 읽기주소 생성기(100)의 동작을 위해 필요한 입력 신호들 중 EP_SIZE와 DEC_CLK을 수신한다. 다음으로, 과정(S300)에서는 읽기주소 생성기(100)의 동작을 위해 필요한 나머지 입력 신호인 DEC_IDX를 수신하고, 읽기주소 생성기(100) 내부의 카운터 값(CNT)을 -1로 초기화(reset)한다. 과정(S400)에서 상기 읽기주소 생성기(100)는 상기 DEC_CLK에 동기하여 상기 카운터 값을 1만큼 증가시키고 이 카운터 값에 대하여 서브블럭 디인터리빙과 서브블럭 심볼 역그룹화를 고려하여 메모리들

(210,220,230) 각각을 위한 칩 선택(Chip Select: CS) 신호들 RAM0_CS, RAM1_CS, RAM2_CS와 읽기 주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR을 생성한다. 과정(S500)에서 메모리부(200)는 상기 칩 선택 신호들 및 상기 읽기 주소들에 따라 메모리들(210,220,230)에서 부호심볼들을 읽어오거나 필요한 경우 제로 삽입(zero insertion)을 수행하여 터보 복호기를 위한 입력 부호심볼들 SYS_DATA, PA0_DATA, PA1_DATA을 출력한다.

<79> 과정(S600)에서 상기 카운터 값을 상기 DEC_CLK에 동기하여 1만큼 증가시킨다. 과정(S700)에서 카운터 값이 상기 EP_SIZE에 해당하는 크기의 부호화 패킷을 처리하기 위한 구성 복호기의 구동 클럭의 발생 회수 즉, N_EP-3보다 작으면 상기 과정(S400)과 상기 과정(S500)을 다시 수행하여 상기 읽기 주소 생성기(100)와 상기 메모리부(200)를 동작시키고, 그렇지 않으면 과정(S800)으로 진행한다. 상기 과정(S800)에서 DEC_END가 '1'이면 한 서브패킷의 부호심볼들이 터보 복호기로 모두 입력된 것으로 판단하여 동작을 종료하고, 그렇지 않으면 상기 과정(S300)으로 회귀하여 DEC_IDX를 다시 수신하고 읽기 주소 생성기(100) 내부의 카운터 값(CNT)을 -1로 리셋(reset)한 후 상기 과정(S400) 내지 상기 과정(S700)을 다시 수행한다.

<80> 도 11은 상기 도 10에 나타낸 상기 과정(S400)에서 상기 읽기 주소 생성기(100)의 동작을 나타낸 흐름도이다. 여기서 나타낸 단계들은 DEC_START가 트리거된 이후 DEC_CLK가 트리거될 때마다 즉, 상기 카운터 값이 1씩 증가할 때마다 수행되는 것이다.

<81> 상기 도 11을 참조하면, 과정(S405)에서 DEC_START가 트리거된 이후 DEC_CLK이 트리거될 때마다 CNT는 1씩 증가한다. CNT는 구성 복호기들 각각으로 입력되는 데이터 비트의 부호심볼(이하 데이터 심볼이라 칭함)을 구분하기 위한 인덱스이다. 부호화 패킷의 크기 N_EP와 각 구성 복호기로 입력되어야 하는 테일비트의 개수 3을 고려하면 CNT는 DEC_CLK가 트리거될 때마다 0부터 N_EP-4(=N_EP-1-3)까지 증가한다.

- <82> 과정(S410)에서는 현재 클럭에서 읽어내야 할 부호심볼이 데이터 비트에 의한 것인지(DT_IDC=0) 또는 테일비트에 의한 것인지(DT_IDC=1)을 결정하기 위하여 CNT를 EP_SIZE-7과 비교한다. 하나의 서브블럭은 한 쌍의 테일비트들(3*2)을 포함하기 때문에 크기가 EP_SIZE-1인 한 서브블럭에서 데이터 비트들의 개수는 EP_SIZE-7(=EP_SIZE-1-3*2)이다. 따라서 CNT가 EP_SIZE-7보다 크면 과정(S415)에서 DT_IDC는 0(테일 비트에 의한 부호심볼)이 되고 그렇지 않으면 과정(S420)에서 DT_IDC는 1(데이터 비트에 의한 부호심볼)이 된다.
- <83> 과정(S425)에서 상기 결정된 DT_IDC와 복호기로부터 입력된 DEC_IDX에 따라 CNT_OS, RAM_SEL, TMP_CS를 생성한다. CNT_OS는 두 번째 구성 복호기(DEC_IDX=1)를 위한 입력 심볼들 중 테일 비트에 의한 부호심볼들(DT_IDC=1)을 구분하기 위해 사용된다(CNT_OS=DT_IDC&&DEC_IDX, 여기서 &&는 AND연산을 의미함.). RAM_SEL은 첫 번째 구성 복호기(DEC_IDX=0)를 위한 입력 심볼들 중 데이터 비트에 의한 부호심볼들(DT_IDC=0)을 읽는 경우에만 0이 된다(RAM_SEL=DT_IDC||DEC_IDX, 여기서 ||는 OR연산자를 의미함.). TMS_CS는 두 번째 구성 복호기(DEC_IDX=1)를 위한 입력 심볼들 중 데이터 비트에 의한 부호심볼들(DT_IDC=0)을 읽는 경우에만 0이 된다.(TMP_CS=DT_IDC||(~DEC_IDX), 여기서 ~는 반전연산자를 의미함.)
- <84> 과정(S430)에서 현재 클럭에서 읽어내야 할 부호심볼이 두 번째 복호기를 위한 테일 비트의 부호심볼(이하 테일 심볼이라 칭함)인지를 결정하기 위해 CNT_OS가 1인지를 판단한다. 도 4와 도 6을 참조하면, t=8, t=9에서 생성된 ENC2(10c)의 테일 심볼들은 t₂=6, t₂=7에서 DEC2(32d)의 입력으로 사용되기 때문에, 이와 같이 두 번째 복호기의 테일 심볼을 읽어내는 경우(CNT_OS=1) 부호심볼 인덱스인 CNT는 테일 비트의 개수만큼 증가되어야 한다. 따라서 만일 CNT_OS가 1이면 과정(S435)에서 새로운 카운터 값 CNTp는 CNT를 3만큼 증가시킨 값이 되고, 그렇지 않으면 과정(S440)에서 CNTp는 CNT와 같게 된다.

- <85> 과정(445)에서는 상기 CNTp에 따라 서브블럭 디인터리빙 규칙에 따른 임시 읽기주소 TMP_ADDR을 생성한다. 여기서 TMP_ADDR은 서브블럭 디인터리빙만을 고려하여 각 서브블럭내에서 읽어내고자 하는 부호심볼의 위치를 나타낸 것으로서 서브블럭 심볼 그룹화는 고려하지 않은 것이다. 따라서 과정(S450)에서는 TMP_ADDR과 앞서 결정한 RAM_SEL을 이용하여 3개의 메모리들 각각을 위한 읽기주소를 생성한다.
- <86> 첫 번째 구성 복호기(DEC_IDX=0)를 위해 데이터 심볼들(DT_IDC=0)이 입력되어야 하는 경우에만(RAM_SEL=0) PA0 심볼 혹은 PA1 심볼이 각각 서브블럭 P0 혹은 서브블럭 P1에 존재하며, 나머지 경우에는 PA0 심볼 혹은 PA1 심볼이 각각 서브블럭 P'0 혹은 서브블럭 P'1에 존재하게 된다. 예를 들어, 도 5에서 데이터 비트에 대한 첫 번째 구성 복호기의 PA0/PA1 심볼만 서브블럭 P0이나 서브블럭 P1에 존재하고 테일 비트에 대한 ENC1의 PA0/PA1 심볼이나 데이터 비트에 대한 ENC2의 PA0/PA1 심볼, 그리고 테일 비트에 대한 ENC2의 PA0/PA1 심볼은 모두 서브블럭 P'0과 P'1 서브블럭에 존재함을 알 수 있다. 서브블럭 심볼 그룹화는 서브블럭 P0의 심볼들과 서브블럭 P'0의 심볼들 혹은 P'0의 심볼들과 서브블럭 P'1의 심볼들을 서로 엇갈리게 배치하기 때문에 이러한 경우 서브블럭 P'0에 속한 심볼들과 서브블럭 P'1에 속한 심볼들이 저장된 주소는 각각 서브블럭 P0에 속한 심볼들과 서브블럭 P'0에 속한 심볼들이 저장된 주소를 2배하여 구할 수 있다. 상기 과정(S450)은 도 13에서 보다 상세히 설명될 것이다.
- <87> 또한 앞서 언급한 도 7에 나타난 바와 같이 사용 가능한 부호심볼들의 종류는 제한되어 있기 때문에 실제로 읽고자 하는 부호심볼이 상기 과정(450)에서 구한 읽기주소에 저장되어 있지 않을 수 있다. 따라서 과정(S460)에서는 TMP_ADDR과 앞서 구한 RAM_SEL 및 TMP_CS를 이용하여 3개의 메모리들을 위한 칩 선택 신호 RAMx_CS를 생성한다. 만일 읽어내고자 하는 부호심볼

이 해당 메모리에 저장되어 있지 않다면 그 메모리의 칩 선택 신호는 0이 되고 대신 복호기로는 제로 심볼이 입력된다. 상기 과정(S460)은 도 15에서 보다 상세히 설명될 것이다.

<88> 도 12는 상기 도 11에 따라 동작하는 상기 읽기주소 생성기(100)의 상세 구성을 나타낸 도면이다.

<89> 상기 도 12를 참조하면, 카운터(105)는 DEC_START가 1이 되는 시점부터 시작하여 DEC_CLK가 트리거될 때마다 CNT를 1씩 증가시킨다. MSB(Most Significant Bit) 추출기(130)는 EP_SIZE에 따라 결정되는 부호화 패킷의 크기 N_EP과 상기 CNT에 따라 (EP_SIZE-7)-CNT의 최상위 비트를 추출하여 DT_IDC로서 출력한다. 앤드 게이트(AND Gate)(135)는 상기 DT_IDC와 DEC_IDX를 앤드하여 CNT_OS를 출력한다. 또한 오아 게이트(OR Gate)(140)는 상기 DT_IDC와 상기 DEC_IDX를 오아하여 RAM_SEL을 출력하고, 오아 게이트(145)는 상기 DT_IDC와 상기 DEC_IDX의 역을 오아하여 TMP_CS를 출력한다.

<90> 선택기(110)는 상기 CNT_OS가 0이면 CNT를 CNTp로서 출력하고 그렇지 않으면 상기 CNT에 가산기(115)에 의해 3이 더해진 CNT+3을 CNTp로서 출력한다. 서브블럭 디인터리버(120)는 상기 CNTp와 EP_SIZE를 이용하여 읽어내고자 하는 부호심볼의 서브블럭내 위치를 나타내는 임시 읽기주소 TMP_ADDR를 생성한다. 상기 서브블럭 디인터리버(120)의 동작은 본 발명의 요지에서 벗어나는 것이므로 그 상세한 설명을 생략하기로 한다.

<91> 그러면 주소 조절기(Address Tuner: ADDR TUNER)(125)는 상기 TMP_ADDR과 상기 EP_SIZE과 상기 RAM_SEL을 이용하여 세 개의 메모리들을 위한 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR를 생성하고, 메모리 선택기(RAM Selector)(150)는 상기 TMP_ADDR과 상기 EP_SIZE과 상기 RAM_SEL과 상기 TMP_CS를 이용하여 세 개의 메모리들을 위한 칩 선택 신호들 RAM0_CS,

RAM1_CS, RAM2_CS를 생성한다. 상기 주소 조절기(125)와 상기 메모리 선택기(150)의 상세한 구성은 도 14와 도 18에 각각 나타내었으며 그 동작에 대한 설명은 후술될 것이다.

<92> 도 13은 상기 도 11에 나타낸 상기 과정(450)에서 상기 주소 조절기(125)에 의한 동작을 나타낸 흐름도이다. 앞서 언급한 도 8을 참조하면, RAM0(210)에는 항상 서브블럭 S의 부호심볼들만이 저장되므로, 모든 경우에 있어서 TMP_ADDR을 RAM0(210)을 위한 읽기 주소 RAM0_ADDR로 사용한다.

<93> 과정(S451)에서 EP_SIZE을 3과 비교한 결과, EP_SIZE이 3보다 작은 경우, 즉 부호화 패킷의 크기 N_EP가 408, 792, 1560인 경우, 도 8을 참조하면 RAM1(220)과 RAM2(230)에는 각각 P0&P'0 그룹의 부호심볼들과 P1&P'1 그룹의 부호심볼들이 저장된다. 그러면 과정(S452)에서 첫 번째 복호기가 데이터 심볼을 요구하는 경우(RAM_SEL=0)인지를 판단한다.

<94> 상술한 바와 같이 RAM_SEL=0인 경우에만 PA0 심볼과 PA1 심볼이 각각 서브블럭 P0과 서브블럭 P1에 존재하며, 나머지 경우에는 PA0 심볼과 PA1 심볼이 각각 서브블럭 P'0과 서브블럭 P'1에 존재한다. 서브블럭 P0과 서브블럭 P1에 속한 부호심볼들은 각각 P0&P'0 그룹의 짝수 위치와 P1&P'1 그룹의 짝수 위치에 존재하며, 서브블럭 P'0과 서브블럭 P'1에 속한 부호심볼들은 각각 P0&P'0 그룹의 홀수 위치와 P1&P'1 그룹의 홀수 위치에 존재한다. 따라서, RAM_SEL이 0이면 과정(S454)에서 RAM0_ADDR은 TMP_ADDR이고, RAM1_ADDR과 RAM2_ADDR은 2*TMP_ADDR로 결정되고, 그렇지 않으면 과정(S453)에서 RAM0_ADDR은 TMP_ADDR이고, RAM1_ADDR과 RAM2_ADDR은 2*TMP_ADDR+1로 결정된다.

<95> 즉, N_EP가 408, 792, 1560인 경우, RAM1(220)을 위한 읽기 주소 RAM1_ADDR과 RAM2(220)을 위한 읽기 주소 RAM2_ADDR은 하기의 <표 1>에 나타낸 바와 같이 RAM_SEL의 값에 따라 결정된다.

<96> 【표 1】

RAM_SEL	RAM1_ADDR	RAM2_ADDR
0	2*TMP_ADDR	2*TMP_ADDR
1	2*TMP_ADDR+1	2*TMP_ADDR+1

<97> 상기 과정(S451)에서 EP_SIZE가 3보다 크거나 같은 경우 과정(S455)로 진행하여 EP_SIZE가 3인지, 즉 N_EP가 2328인지를 판단한다. 이 경우 RAM1(220)에는 서브블럭 P0의 부호심볼들 전체와 서브블럭 P'1의 부호심볼들의 일부(408개)가 존재하며, RAM2(230)에는 서브블럭 P'0의 부호심볼들 전체와 서브블럭 P1의 부호심볼들의 일부(408개)가 존재한다. 이 경우는 N_EP가 408, 792, 1560인 경우와 달리 P0&P'0 그룹과 P1&P'1 그룹의 부호심볼들은 이미 서브블럭 심볼 역그룹화가 고려되어 분리된 채 RAM1(220)과 RAM2(230)에 저장되어 있다. 이는 RAM1(220)과 RAM2(230)의 크기가 3120으로 제한되어 있기 때문이다.

<98> EP_SIZE가 3인 경우 과정(S457)에서 첫 번째 복호기가 데이터 심볼을 요구하는 경우 (RAM_SEL=0)인지를 판단한다. N_EP가 2328인 경우에도 RAM_SEL=0인 경우에만 PA0 심볼과 PA1 심볼이 각각 서브블럭 P0과 서브블럭 P1에 존재하며, 나머지 경우에는 PA0 심볼과 PA1 심볼이 각각 서브블럭 P'0과 서브블럭 P'1에 존재한다. 이 때, 서브블럭 P'1의 부호심볼들은 RAM1(220)의 뒷부분 즉, 주소가 2328부터 2735(=2328+408-1)인 영역에 위치해 있고, 서브블럭 P1의 부호심볼들은 RAM2(230)의 뒷부분 즉, 주소가 2328부터 2735(=2328+408-1)인 영역에 위치해 있다. 따라서, RAM_SEL이 0이면 과정(S458)에서 RAM0_ADDR과 RAM1_ADDR은 TMP_ADDR이고 RAM2_ADDR은 TMP_ADDR+2328이며, 그렇지 않으면 과정(S459)에서 RAM0_ADDR과 RAM2_ADDR은 TMP_ADDR이고 RAM1_ADDR은 TMP_ADDR+2328로 결정된다.

<99> 즉, N_EP가 2328인 경우, RAM1(220)을 위한 읽기 주소 RAM1_ADDR과 RAM2(230)를 위한 읽기 주소 RAM2_ADDR은 하기의 <표 2>에 나타낸 바와 같이 RAM_SEL의 값에 따라 결정된다.

<100> 【표 2】

RAM_SEL	RAM1_ADDR	RAM2_ADDR
0	TMP_ADDR	TMP_ADDR+2328
1	TMP_ADDR+2328	TMP_ADDR

<101> 마지막으로 상기 과정(S455)에서 EP_SIZE가 3보다 큰 것으로 판단된 경우 즉, N_EP가 3096, 3864인 경우에는 서브블럭 P1과 서브블럭 P'1의 부호심볼들이 메모리에 아예 존재하지 않으며, RAM1(220)에는 서브블럭 P0의 부호심볼들만이 저장되고, RAM2(230)에는 서브블럭 P'0의 부호심볼들만이 저장된다. 따라서, 과정(S456)에서 RAM0_ADDR과 RAM1_ADDR과 RAM2_ADDR은 모두 TMP_ADDR로 결정된다.

<102> 즉, EP_SIZE은 3096, 3864인 경우, RAM1(220)을 위한 읽기 주소 RAM1_ADDR과 RAM2(230)를 위한 읽기 주소 RAM2_ADDR은 하기의 <표 3>에 나타낸 바와 같이 RAM_SEL의 값과 상관없이 결정된다.

<103> 【표 3】

RAM_SEL	RAM1_ADDR	RAM2_ADDR
0	TMP_ADDR	TMP_ADDR
1	TMP_ADDR	TMP_ADDR

<104> 도 14a, 14b, 14c는 상기 도 13에 따라 동작하는 상기 주소 조절기(125)의 상세 구성을 나타낸 도면이다.

<105> 상기 도 14a를 참조하여 상기 주소 조절기(125)에서 N_EP가 408, 792, 1560인 경우를 위한 부분(125a)에 대해 설명하면, TMP_ADDR은 그대로 RAM0_ADDR로서 출력되며, 곱셈기(70)는

TMP_ADDR에 2를 곱하고 가산기(71)는 상기 곱셈기(70)의 출력에 RAM_SEL을 더하여 RAM1_ADDR 및 RAM2_ADDR로서 출력한다. 상기 도 14b를 참조하여 N_EP가 2328인 경우를 위한 부분(125b)에 대해 설명하면, TMP_ADDR은 그대로 RAM0_ADDR로서 출력되며, 선택기(72)는 RAM_SEL이 1인 경우에만 2328을 출력하며 선택기(74)는 RAM_SEL이 0인 경우에만 2328을 출력한다. 그러면 덧셈기(73)은 TMP_ADDR에 상기 선택기(72)의 출력(2328 또는 0)을 더하여 RAM1_ADDR로서 출력하고, 덧셈기(75)는 TMP_ADDR에 상기 선택기(74)의 출력(0 또는 2328)을 더하여 RAM2_ADDR로서 출력한다. 상기 도 14c를 참조하여 N_EP가 3096, 3864인 경우를 위한 부분(125c)에 대해 설명하면, TMP는 그대로 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR로서 출력된다.

<106> 도 15는 상기 도 11에 나타낸 상기 과정(S460)에서 상기 메모리 선택기(150)에 의한 동작을 나타낸 흐름도이다. 상기 메모리 선택기(150)는 세 개의 RAM에 대한 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS를 생성하는데, 읽어들 부호심볼이 없는 RAM의 칩 선택 신호를 0으로 설정한다. 칩 선택 신호가 0으로 설정되어 있으면 해당 RAM을 액세스하기 위한 클럭이 공급되지 않으므로 RAM에 있는 부호심볼들을 읽어들 수 없다.

<107> 앞서 언급한 도 8에 나타낸 바에 따르면, EP_SIZE에 상관 없이 RAM0(210)에 저장되어 있는 서브블럭 S의 부호심볼들을 읽을 필요가 없는 공통적인 경우는 두 번째 구성 복호기(DEC_IDX=1)를 위해 데이터 비트의 부호심볼들(DT_IDC=0)을 읽어내는 경우이다. 도 6에 나타낸 바와 같이 이 경우에는 터보 인터리빙된 첫 번째 구성 복호기의 출력 데이터가 두 번째 구성 복호기의 입력으로 사용될 것이므로 RAM0(210)을 액세스할 필요가 없다. 이는 도 11에 나타낸 TMP_CS에 의해 판단할 수 있다. 즉, 하기의 <표 4>에 나타낸 바와 같이 TMP_CS는 EP_SIZE에 상관없이 DT_IDC가 0이고 DEC_IDX가 1인 경우에만 0이고 나머지 경우에 항상 1이다.

<108> 【표 4】

DT_IDC (A)	DEC_IDX (B)	TMP_CS == OR(A, ~B)
0	0	1
	1	0
1	0	1
	1	1

<109> 따라서, 과정(S461)에서 RAM0_CS는 TMP_CS의 값을 그대로 사용한다.

<110> 과정(S462)에서는 RAM1_CS와 RAM2_CS를 결정하기 위하여 EP_SIZE가 3보다 작은지를 판단한다. EP_SIZE가 3보다 작은 경우 즉, N_EP가 408, 792, 1560인 경우 RAM1(220)와 RAM2(230)는 각각 P0&P'0 그룹과 P1&P'1 그룹의 모든 부호심볼들을 저장하고 있으므로 과정(S463)에서 RAM1_CS와 RAM2_CS는 항상 1이다. 과정(S464)에서는 EP_SIZE가 3과 같은지를 판단하여 만일 같으면 과정(S465)로 진행하고, 같지 않으면 과정(S466)로 진행한다.

<111> 도 16은 N_EP가 2328인 경우(EP_SIZE=3) 상기 도 15에 나타난 상기 과정(465)에서 상기 메모리 선택기(150)가 RAM1_CS와 RAM2_CS를 결정하는 동작을 나타낸 흐름도이다.

<112> 앞서 언급한 도 8을 참조하여 상기 도 16을 설명하면, 과정(S10)에서의 판단결과 TMP_ADDR이 408 이상인 경우, RAM1(220)에는 서브블럭 P'1의 부호심볼들이 더 이상 저장될 수 없고 RAM2(230)에는 서브블럭 P1의 부호심볼들이 더 이상 저장될 수 없다. 따라서, TMP_ADDR이 408이상이면 과정(S12)에서 변수 TH_OUT를 1로 결정하고 그렇지 않으면 TH_OUT를 0으로 설정한다. 여기서 TH_OUT는 TMP_ADDR이 정해진 임계값(408)을 초과하는지를 판단하는 기준이다.

<113> 서브블럭 P'1의 부호심볼들이 요구되는(즉 읽어내야 하는) 경우는 RAM_SEL=1인 경우이며, 서브블럭 P1에 속하는 부호심볼들이 요구되는 경우는 RAM_SEL=0인 경우이다. 따라서 과정(S13)에서의 판단결과 RAM_SEL이 1이고 TH_OUT가 1이면 (S15)로 진행하여 RAM1_CS는 0이 되며, 그렇지 않으면 과정(S14)로 진행하여 RAM1_CS는 1이 된다. 또한 과정(S16)에서의 판단결

과 RAM_SEL이 0이고 TH_OUT가 1이면 (S17)로 진행하여 RAM2_CS는 0이 되고 그렇지 않으면
과정(S18)로 진행하여 RAM2_CS는 1이 된다.

<114> 즉, N_EP가 2328인 경우 RAM1_CS와 RAM2_CS는 하기의 <표 5>와 같이 RAM_SEL과 TH_OUT에
의해 결정된다.

<115> 【표 5】

RAM_SEL (A)	TH_OUT (B)	RAM1_CS == NAND(A, B)	RAM2_CS == NAND(~A, B)
0	0	1	1
	1	1	0
1	0	1	1
	1	0	1

<116> 도 17은 N_EP가 3096, 3864인 경우(EP_SIZE>3) 상기 도 15에 나타낸 상기 과정(466)에서
상기 메모리 선택기(150)가 RAM1_CS와 RAM2_CS를 결정하는 동작을 나타낸 흐름도이다.

<117> 상기 도 17을 참조하면, 과정(S20)에서 N_EP이 4인지(EP_SIZE가 3096인지)를 판단한다.
EP_SIZE가 3096인 경우 과정(S21)에서 TMP_ADDR이 2352 이상인지를 판단한다. 앞서 언급한 도
8을 참조하면, TMP_ADDR이 2352 이상이면 RAM1(220)과 RAM2(230)는 각각 서브블럭 P0와 서브블
럭 P'0의 부호심볼들을 더 이상 저장하지 못한다. 따라서 TMP_ADDR이 2352 이상이면 과정(S22)
로 진행하여 TH_OUT를 1로 결정하고, 그렇지 않으면 과정(S24)로 진행하여 TH_OUT를 0으로 결
정한다.

<118> N_EP가 3864인 경우 마찬가지로 과정(S23)에서 TMP_ADDR이 1968 이상인지를 판단한다.
TMP_ADDR이 1968 이상이면 RAM1(220)과 RAM2(230)는 각각 서브블럭 P0와 서브블럭 P'0의 부호
심볼들을 더 이상 저장하지 못한다. 따라서 TMP_ADDR이 1968 이상이면 과정(S22)로 진행하여
TH_OUT를 1로 결정하고, 그렇지 않으면 과정(S24)로 진행하여 TH_OUT를 0으로 결정한다. 마찬

가지로 여기서 TH_OUT는 TMP_ADDR이 정해진 임계값(2352 또는 1968)을 초과하는지를 판단하는 기준이다.

<119> 서브블럭 P0에 속하는 부호심볼들이 요구되는(즉 읽어내야 하는) 경우는 RAM_SEL=0인 경우이며, 서브블럭 P'0에 속하는 부호심볼들이 요구되는 경우는 RAM_SEL=1인 경우이다. 따라서 과정(S25)에서의 판단결과 RAM_SEL이 0이고 TH_OUT가 0이면 과정(S27)로 진행하여 RAM1_CS를 1로 결정하고 그렇지 않으면 과정(S26)으로 진행하여 RAM1_CS를 0으로 결정한다. 또한 과정(S28)에서의 판단결과 RAM_SEL이 1이고 TH_OUT가 0이면 과정(S29)로 진행하여 RAM2_CS를 1로 결정하고 그렇지 않으면 과정(S30)으로 진행하여 RAM2_CS를 0으로 결정한다.

<120> 즉, N_EP가 3096, 3864인 경우 RAM1_CS와 RAM2_CS는 하기의 <표 6>과 같이 RAM_SEL과 TH_OUT에 의해 결정된다.

<121> 【표 6】

RAM_SEL (A)	TH_OUT (B)	RAM1_CS == NOR(A, B)	RAM2_CS == NOR(~A, B)
0	0	1	0
	1	0	0
1	0	0	1
	1	0	0

<122> 도 18a, 18b, 18c, 18d는 상기 도 15 내지 도 17에 따라 동작하는 상기 메모리 선택기(150)의 상세 구성을 나타낸 도면이다.

<123> 상기 도 18a를 참조하여 상기 메모리 선택기(150)에서 N_EP가 408, 792, 1560인 경우를 위한 부분(150a)에 대해 설명하면, TMP_CS는 그대로 RAM0_CS가 되며, RAM1_CS와 RAM2_CS는 항상 1이다. 상기 도 18b를 참조하여 상기 메모리 선택기(150)에서 N_EP가 2328인 경우를 위한 부분(150b)에 대해 설명하면, TMP_CS는 그대로 RAM0_CS가 된다. MSB 추출기(151)가 TMP_ADDR을

입력으로 하여 (407-TMP_ADDR)의 최상위비트인 TH_OUT를 추출하면 난드 게이트(NAND Gate)(152)는 RAM_SEL과 TH_OUT를 난드하여 RAM1_CS를 출력하며, 난드 게이트(153)는 RAM_SEL의 역과 TH_OUT를 난드하여 RAM2_CS를 출력한다. 상기 도 18c를 참조하여 상기 메모리 선택기(150)에서 N_EP가 3096인 경우를 위한 부분(150c)에 대해 설명하면, TMP_CS는 그대로 RAM0_CS가 된다. MSB 추출기(154)가 TMP_ADDR를 입력으로 하여 (2351-TMP_ADDR)의 최상위비트인 TH_OUT를 추출하면 노아 게이트(NOR Gate)(155)는 RAM_SEL과 TH_OUT를 노아하여 RAM1_CS를 출력하며, 노아 게이트(156)는 RAM_SEL의 역과 TH_OUT를 난드하여 RAM2_CS를 출력한다. 상기 도 18d를 참조하여 상기 메모리 선택기(150)에서 N_EP가 3864인 경우를 위한 부분(150d)에 대해 설명하면, TMP_CS는 그대로 RAM0_CS가 된다. MSB 추출기(157)가 TMP_ADDR를 입력으로 하여 (1967-TMP_ADDR)의 최상위비트인 TH_OUT를 추출하면 노아 게이트(NOR Gate)(158)는 RAM_SEL과 TH_OUT를 노아하여 RAM1_CS를 출력하며, 노아 게이트(159)는 RAM_SEL의 역과 TH_OUT를 난드하여 RAM2_CS를 출력한다.

<124> 도 19는 상기 도 10에 나타난 상기 과정(S500)에서 상기 메모리부(200)의 동작을 나타낸 흐름도이다. 여기서 상기 메모리부(200)는 상기 읽기주소 생성기(100)으로부터의 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR 및 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS에 응답하여 메모리들(210, 220, 230)에서 읽어낸 부호심볼들을 출력하고 필요한 경우 출력된 부호심볼들을 상호 교환(switch)하거나, 읽어낸 부호심볼이 없는 경우 대신 제로 심볼을 출력한다.

<125> 상기 도 19를 참조하면, 과정(S505)에서의 판단결과 RAM0_CS가 1이면 과정(S515)로 진행하여 RAM0의 RAM0_ADDR에서 읽어낸 RAM0_DATA를 SYS_DATA로서 출력하고, 그렇지 않은 경우 과정(S510)으로 진행하여 제로 심볼을 SYS_DATA로서 출력한다. 과정(S520)에서의 판단결과

RAM1_CS가 1이면 과정(S530)로 진행하여 RAM1의 RAM1_ADDR에서 읽어낸 RAM1_DATA를 PA0_TMP로서 출력하고, 그렇지 않은 경우 과정(S525)로 진행하여 제로 심볼을 PA0_TMP로서 출력한다. 또한 과정(S555)에서의 판단결과 RAM2_CS가 1이면 과정(S560)로 진행하여 RAM2_ADDR에서 읽어낸 RAM2_DATA를 PA1_TMP로서 출력하고 그렇지 않은 경우 과정(S565)로 진행하여 제로 심볼을 PA1_TMP로서 출력한다.

<126> 앞서 언급한 도 8을 참조하면, EP_SIZE가 3보다 작은 경우 즉, N_EP가 408, 792, 1560인 경우에는 RAM1(220)에서 읽어낸 부호심볼은 항상 서브블럭 P0 혹은 서브블럭 P'0에 속하고 RAM2(230)에서 읽어낸 부호심볼은 항상 서브블럭 P1 혹은 서브블럭 P'1에 속한다. 그런데 EP_SIZE가 3 즉, N_EP가 2328인 경우에는 TMP_ADDR이 2328 이상의 메모리 영역에서 서브블럭 P'1의 부호심볼이 RAM1(220)에 저장되어 있고 서브블럭 P'0의 부호심볼이 RAM2(230)에 저장되어 있다. EP_SIZE가 3보다 큰 경우 즉, N_EP가 3096이나 3864인 경우에도 서브블럭 P'0의 부호심볼이 RAM2(230)에 저장되어 있다. 따라서 서브블럭 P0과 서브블럭 P1의 부호심볼이 첫 번째 구성 복호기로 입력되고 서브블럭 P'0과 서브블럭 P'1의 부호심볼이 두 번째 구성 복호기로 입력되도록 하기 위해서는, 이러한 경우들을 식별하여 RAM1(220)에서 읽어낸 부호심볼과 RAM2(230)에서 읽어낸 부호심볼을 상호 교환하여야 한다.

<127> 과정(S535)에서 EP_SIZE가 3보다 작은 것으로 판단된 경우, 과정(S540)으로 진행하여 RAM1(220)에서 읽어낸 PA0_TMP와 RAM2(230)에서 읽어낸 PA1_TMP는 교환없이 각각 PA0_DATA와 PA1_DATA가 된다. EP_SIZE가 3보다 작지 않은 경우 과정(S545)에서 RAM_SEL이 0인지를 판단한다. 이는 서브블럭 P0 혹은 서브블럭 P1의 부호심볼이 요구되었는지(RAM_SEL=0) 아니면 서브블럭 P'0 혹은 서브블럭 P'1의 부호심볼이 요구되었는지(RAM_SEL=1)에 따라 RAM1(220)의 부호심볼과 RAM2(230)의 부호심볼을 교환할지의 여부를 결정할 수 있기 때문이다. 즉 상기 과정

(S545)에서 RAM_SEL이 0이 아니면 과정(S550)으로 진행하여 RAM1(220)에서 읽어낸 PA0_TMP와 RAM2(230)에서 읽어낸 PA1_TMP는 상호 교환에 의해 각각 PA1_DATA와 PA0_DATA가 되고, 그렇지 않으면 상기 과정(S540)으로 진행하여 교환을 수행하지 않는다.

<128> 도 20은 상기 도 19에 따라 동작하는 상기 메모리부(200)의 상세 구성을 나타낸 도면이다. 도시된 바와 같이 상기 메모리부(200)는 3개의 RAM(210,220,230)과 제로 삽입기(240)와 RAM 스위치(260)로 구분되며,

<129> 상기 읽기주소 생성기(100)으로부터의 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR와 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS 및 RAM_SEL, EP_SIZE에 응답하여, 시스템틱 부호 심볼 SYS_DATA과 패리티 부호심볼들 PA0_DATA, PA1_DATA를 동시에 출력한다.

<130> 상기 RAM(210,220,230)는 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR에 응답하여 RAM0_DATA, RAM1_DATA, RAM2_DATA를 출력한다. 상기 제로 삽입기(240)는 RAM0_CS, RAM1_CS, RAM2_CS에 따라 각각 SYS_DATA, PA0_TMP, PA1_TMP를 결정하기 위하여, 3개의 선택기들(215,225,235)로 구성된다. 선택기(215)는 RAM0_CS가 1인 경우에만 상기 RAM0(210)으로부터의 RAM0_DATA를 선택하고 RAM0_CS가 0인 경우에는 제로 심볼을 선택하여 출력한다. 선택기(225)는 RAM1_CS가 1인 경우에만 상기 RAM1(220)으로부터의 RAM1_DATA를 선택하고 RAM1_CS가 0인 경우에는 제로 심볼을 선택하여 PA0_TMP로서 출력한다. 선택기(235)는 RAM2_CS가 1인 경우에만 상기 RAM2(230)으로부터의 RAM2_DATA를 선택하고 RAM2_CS가 0인 경우에는 제로 심볼을 선택하여 PA1_TMP로서 출력한다.

<131> 상기 RAM 스위치(260)는 PA0_TMP와 PA1_TMP를 PA0_DATA와 PA1_DATA에 적절히 대응시키기 위하여 3개의 선택기들(245,250,255)로 구성된다. EP_SIZE가 3보다 작은 경우 즉, N_EP가 408,792,1560인 경우 선택기(255)는 0을 선택하여

선택기들(245,250)의 선택신호로서 제공한다. 그러면 선택기들(245,250)는 PA0_TMP와 PA1_TMP를 그대로 PA0_DATA와 PA1_DATA로서 출력한다. 반면 EP_SIZE가 3보다 작지 않은 경우 즉 N_EP가 2328,3096,3864인 경우 선택기(255)는 RAM_SEL을 선택하여 선택기들(245,250)의 선택신호로서 제공한다. 그러면 선택기들(245,250)은 상기 RAM_SEL이 0인 경우에는 PA0_TMP와 PA1_TMP를 그대로 PA0_DATA와 PA1_DATA로서 출력하고, 그렇지 않은 경우에는 PA0_TMP와 PA1_TMP를 상호 교환하여 PA1_DATA와 PA0_DATA로서 출력한다.

【발명의 효과】

- <132> 이상에서 상세히 설명한 바와 같이 동작하는 본 발명에 있어서, 개시되는 발명 중 대표적인 것에 의하여 얻어지는 효과를 간단히 설명하면 다음과 같다.
- <133> 본 발명은 1xEV-DV 표준에 따른 단말기에서 채널 인터리빙된 데이터를 고속으로 처리하여 복호기로 입력함으로써 필요한 버퍼의 개수를 줄여 소형화, 저가격화, 저전력화를 실현하며, 지연을 감소시켜 고속의 데이터 서비스를 지원할 수 있도록 한다.

【특허청구범위】

【청구항 1】

송신기에서 소정 개수의 데이터 비트들과 테일 비트들을 포함하는 부호화 패킷을 소정 부호율에 따라 부호화하여 생성한 5개의 서브블럭들의 부호심볼들을 인터리빙하여 전송하면, 수신기에서 상기 부호심볼들을 수신하여 저장하고 상기 인터리빙에 대응하는 디인터리빙 순서대로 읽어내어 터보 복호기로 입력하는 방법에 있어서,

상기 수신된 부호심볼들은 종류에 따라 3개의 메모리들 RAM0, RAM1, RAM2에 나누어 저장되고, 상기 터보 복호기는 순차적으로 동작하면서 상기 3개의 메모리들로부터 읽어낸 3개의 부호심볼들을 복호하는 2개의 구성 복호기들로 구성되며,

상기 터보 복호기로부터 수신되는 부호화 패킷의 크기와 복호기 클럭과 복호기 인덱스 DEC_IDX를 이용하여 상기 3개의 메모리들을 위한 읽기주소들 RAMx_ADDR과 칩 선택 신호들 RAMx_CS를 생성하는 과정과,

상기 생성된 읽기주소들 RAMx_ADDR과 상기 칩 선택 신호들 RAMx_CS를 이용하여 상기 2개의 구성 복호기들 중 하나로 입력될 3개의 부호심볼들을 출력하는 과정을 포함하는 것을 특징으로 하는 상기 방법.

【청구항 2】

제 1 항에 있어서, 상기 읽기주소들과 칩 선택 신호들을 선택하는 과정은,

상기 복호기 클럭이 인가될 때마다 상기 터보 복호기로 데이터 심볼을 입력하는 경우 DT_IDC를 0으로 설정하고 테일 심볼들을 입력하는 경우 상기 DT_IDC를 1로 설정하는 단계와,

상기 터보 복호기로 입력하고자 하는 부호심볼의 해당하는 서브블럭내의 인터리빙된 위치를 나타내는 TMP_ADDR을 생성하는 단계와,

상기 DT_IDC와 상기 DEC_IDX를 오아하여 생성한 RAM_SEL과 상기 TMP_ADDR와 상기 부호화 패킷의 크기를 이용하여 상기 읽기주소들 RAMx_ADDR을 생성하는 단계와,

상기 DT_IDC와 상기 DEC_IDX의 역을 오아하여 생성한 TMP_CS와 상기 RAM_SEL과 상기 TMP_ADDR와 상기 부호화 패킷의 크기를 이용하여 상기 칩 선택 신호들 RAMx_CS을 생성하는 단계를 포함하는 것을 특징으로 하는 상기 방법.

【청구항 3】

제 2 항에 있어서, 상기 부호화 패킷의 크기가 408,792,1560 중 하나인 경우, 상기 읽기 주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 하기의 <수학식 1>에 따라 결정되는 것을 특징으로 하는 상기 방법.

```

if RAM_SEL=0
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=2*TMP_ADDR
    RAM2_ADDR=2*TMP_ADDR
else
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=2*TMP_ADDR+1
    RAM2_ADDR=2*TMP_ADDR+1

```

【수학식 1】

【청구항 4】

제 3 항에 있어서, 상기 칩 선택 신호 RAM0_CS은 상기 TMP_CS이고, 상기 RAM1_CS, RAM2_CS는 1로 결정되는 것을 특징으로 하는 상기 방법.

【청구항 5】

제 2 항에 있어서, 상기 부호화 패킷의 크기가 2328인 경우, 상기 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 하기의 <수학식 2>에 따라 결정되는 것을 특징으로 하는 상기 방법.

```

if RAM_SEL=0
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=TMP_ADDR
    RAM2_ADDR=TMP_ADDR+2328
else
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=TMP_ADDR+2328

```

【수학식 2】 $RAM2_ADDR=TMP_ADDR$

【청구항 6】

제 5 항에 있어서, 상기 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS는 하기의 <수학식 3>에 따라 결정되는 것을 특징으로 하는 상기 방법.

```

if (RAM_SEL=1)&&(TMP_ADDR>=408)
    RAM1_CS=0
else
    RAM1_CS=1
if (RAM_SEL=0)&&(TMP_ADDR>=408)
    RAM2_CS=0
else
    RAM2_CS=1

```

【수학식 3】

【청구항 7】

제 2 항에 있어서, 상기 부호화 패킷의 크기가 3096 또는 3864인 경우, 상기 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 모두 상기 TMP_ADDR로 결정되는 것을 특징으로 하는 상기 방법.

【청구항 8】

제 7 항에 있어서, 상기 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS는 하기의 <수학식 4>에 따라 결정되며, 하기의 <수학식 4>에서 TH_OUT은 상기 TMP_ADDR이 소정의 임계값 이상일 때 1이고 그렇지 않을 경우 0이고, 상기 소정의 임계값은 상기 부호화 패킷의 크기가 3096인 경우 2352이고 상기 부호화 패킷의 크기가 3864인 경우 1968인 것을 특징으로 하는 상기 방법.

```

if (RAM_SEL=0)&&(TH_OUT=0)
    RAM1_CS=1
else
    RAM1_CS=0
if (RAM_SEL=1)&&(TH_OUT=0)
    RAM2_CS=1
else

```

【수학식 4】 RAM2_CS=0

【청구항 9】

제 1 항에 있어서, 상기 부호화 패킷의 크기가 408,792,1560 중 하나이거나 또는 상기 부호화 패킷의 크기가 2328,3096,3864 중 하나이면서 상기 2개의 구성 복호기들 중 첫 번째 구성 복호기로 데이터 비트에 의한 부호심볼이 입력되는 경우, 상기 출력되는 3개의 부호심볼들 SYS_DATA, PA0_DATA, PA1_DATA는, 하기의 <수학식 5>에 따라 결정되는 것을 특징으로 하는 상기 방법.

```

if RAM0_CS=1
    SYS_DATA=RAM0_DATA
else
    SYS_DATA=0
if RAM1_CS=1
    PA0_DATA=RAM1_DATA
else
    PA0_DATA=0
if RAM2_CS=1
    PA1_DATA=RAM2_DATA
else

```

【수학식 5】 PA1_DATA=0

【청구항 10】

제 1 항에 있어서, 상기 부호화 패킷의 크기가 2328,3096,3864 중 하나이면서 상기 2개의 구성 복호기들 중 두 번째 구성 복호기로 부호심볼이 입력되거나 첫 번째 구성 복호기로 테일비트에 의한 부호심볼이 입력되는 경우, 상기 출력되는 3개의 부호심볼들 SYS_DATA, PA0_DATA, PA1_DATA는, 하기의 <수학식 6>에 따라 결정되는 것을 특징으로 하는 상기 방법.

```

if RAM0_CS=1
    SYS_DATA=RAM0_DATA
else
    SYS_DATA=0
if RAM1_CS=1
    PA0_DATA=RAM2_DATA
else
    PA0_DATA=0
if RAM2_CS=1
    PA1_DATA=RAM1_DATA
else

```

【수학식 6】 PA1_DATA=0

【청구항 11】

송신기에서 소정 개수의 데이터 비트들과 테일 비트들을 포함하는 부호화 패킷을 소정 부호율에 따라 부호화하여 생성한 5개의 서브블럭들의 부호심볼들을 인터리빙하여 전송하면, 수신기에서 상기 부호심볼들을 수신하여 저장하고 상기 인터리빙에 대응하는 디인터리빙 순서대로 읽어내어 터보 복호기로 입력하는 장치에 있어서,

상기 수신된 부호심볼들은 종류에 따라 3개의 메모리들 RAM0, RAM1, RAM2에 나누어 저장되고, 상기 터보 복호기는 순차적으로 동작하면서 상기 3개의 메모리들로부터 읽어낸 3개의 부호심볼들을 복호하는 2개의 구성 복호기들로 구성되며,

상기 터보 복호기로부터 수신되는 부호화 패킷의 크기와 복호기 클럭과 복호기 인덱스 DEC_IDX를 이용하여 상기 3개의 메모리들을 위한 읽기주소들 RAMx_ADDR과 칩 선택 신호들 RAMx_CS를 생성하는 입력주소 생성기와,

상기 생성된 읽기주소들 RAMx_ADDR과 상기 칩 선택 신호들 RAMx_CS를 이용하여 상기 2개의 구성 복호기들 중 하나로 입력될 3개의 부호심볼들을 출력하는 메모리부를 포함하는 것을 특징으로 하는 상기 장치.

【청구항 12】

제 11 항에 있어서, 상기 읽기주소 생성기는,

-1 부터 시작하여 상기 복호기 클럭이 인가될 때마다 1씩 증가하는 CNT를 출력하는 카운터와,

상기 CNT에 3을 더하는 가산기와,

"상기 부호화 패킷의 크기-7-상기 CNT"의 최상위 비트를 추출하여 DT_IDC를 출력하는 최상위비트 추출기와,

상기 DT_IDC와 상기 DEC_IDX를 앤드하여 CNT_OS를 출력하는 앤드 게이트와,

상기 DT_IDC와 상기 DEC_IDX를 오아하여 RAM_SEL을 출력하는 제1 오아 게이트와,

상기 DT_IDC와 상기 DEC_IDX의 역을 오아하여 TMP_CS를 출력하는 제2 오아 게이트와,

상기 CNT_OS가 0이면 상기 CNT를 선택하고 상기 CNT_OS가 1이면 상기 CNT+3을 선택하여 CNTp를 출력하는 선택기와,

상기 CNTp를 이용하여, 상기 터보 복호기로 입력하고자 하는 부호심볼에 해당하는 서브블럭내의 인터리빙된 위치를 나타내는 TMP_ADDR을 생성하는 서브블럭 디인터리버와,

상기 TMP_ADDR와 상기 부호화 패킷의 크기와 상기 RAM_SEL을 이용하여 상기 읽기주소들 RAMx_ADDR을 생성하는 주소 조절기와,

상기 TMP_ADDR와 상기 부호화 패킷의 크기와 상기 RAM_SEL과 상기 TMP_CS를 이용하여 상기 칩 선택 신호들 RAMx_CS를 생성하는 칩 선택 신호 생성기로 구성되는 것을 특징으로 하는 상기 장치.

【청구항 13】

제 12 항에 있어서, 상기 부호화 패킷의 크기가 408,792,1560 중 하나인 경우, 상기 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 하기의 <수학식 7>에 따라 결정되는 것을 특징으로 하는 상기 장치.

```

if RAM_SEL=0
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=2*TMP_ADDR
    RAM2_ADDR=2*TMP_ADDR
else
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=2*TMP_ADDR+1
    RAM2_ADDR=2*TMP_ADDR+1

```

【수학식 7】

【청구항 14】

제 13 항에 있어서, 상기 칩 선택 신호 RAM0_CS은 상기 TMP_CS이고, 상기 RAM1_CS, RAM2_CS는 1로 결정되는 것을 특징으로 하는 상기 장치.

【청구항 15】

제 12 항에 있어서, 상기 부호화 패킷의 크기가 2328인 경우, 상기 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 하기의 <수학식 8>에 따라 결정되는 것을 특징으로 하는 상기 장치.

```

if RAM_SEL=0
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=TMP_ADDR
    RAM2_ADDR=TMP_ADDR+2328
else
    RAM0_ADDR=TMP_ADDR
    RAM1_ADDR=TMP_ADDR+2328
    RAM2_ADDR=TMP_ADDR

```

【수학식 8】

【청구항 16】

제 15 항에 있어서, 상기 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS는 하기의 <수학식 9>에 따라 결정되는 것을 특징으로 하는 상기 장치.

```

if (RAM_SEL=1)&&(TMP_ADDR.>=408)
    RAM1_CS=0
else
    RAM1_CS=1
if (RAM_SEL=0)&&(TMP_ADDR.>=408)
    RAM2_CS=0
else

```

【수학식 9】 RAM1_CS=1

【청구항 17】

제 12 항에 있어서, 상기 부호화 패킷의 크기가 3096 또는 3864인 경우, 상기 읽기주소들 RAM0_ADDR, RAM1_ADDR, RAM2_ADDR은 모두 상기 TMP_ADDR로 결정되는 것을 특징으로 하는 상기 장치.

【청구항 18】

제 17 항에 있어서, 상기 칩 선택 신호들 RAM0_CS, RAM1_CS, RAM2_CS는 하기의 <수학식 10>에 따라 결정되며, 하기의 <수학식 10>에서 TH_OUT은 상기 TMP_ADDR이 소정의 임계값 이상일 때 1이고 그렇지 않을 경우 0이고, 상기 소정의 임계값은 상기 부호화 패킷의 크기가 3096인 경우 2352이고 상기 부호화 패킷의 크기가 3864인 경우 1968인 것을 특징으로 하는 상기 장치.

```

if (RAM_SEL=0)&&(TH_OUT=0)
    RAM1_CS=1
else
    RAM1_CS=0
if (RAM_SEL=1)&&(TH_OUT=0)
    RAM2_CS=1
else

```

【수학식 10】 RAM2_CS=0

【청구항 19】

제 11 항에 있어서, 상기 부호화 패킷의 크기가 408,792,1560 중 하나이거나 또는 상기 부호화 패킷의 크기가 2328,3096,3864 중 하나이면서 상기 2개의 구성 복호기들 중 첫 번째 구성 복호기로 데이터 비트에 의한 부호심볼이 입력되는 경우, 상기 메모리부로부터 출력되는 3개의 부호심볼들 SYS_DATA, PA0_DATA, PA1_DATA는, 하기의 <수학식 11>에 따라 결정되는 것을 특징으로 하는 상기 장치.

```

if RAM0_CS=1
    SYS_DATA=RAM0_DATA
else
    SYS_DATA=0
if RAM1_CS=1
    PA0_DATA=RAM1_DATA
else
    PA0_DATA=0
if RAM2_CS=1
    PA1_DATA=RAM2_DATA
else

```

【수학식 11】 PA1_DATA=0

【청구항 20】

제 11 항에 있어서, 상기 부호화 패킷의 크기가 2328,3096,3864 중 하나이면서 상기 2개의 구성 복호기들 중 두 번째 구성 복호기로 부호심볼이 입력되거나 첫 번째 구성 복호기로 테일비트에 의한 부호심볼이 입력되는 경우, 상기 출력되는 3개의 부호심볼들 SYS_DATA, PA0_DATA, PA1_DATA는, 하기의 <수학식 12>에 따라 결정되는 것을 특징으로 하는 상기 장치.

```

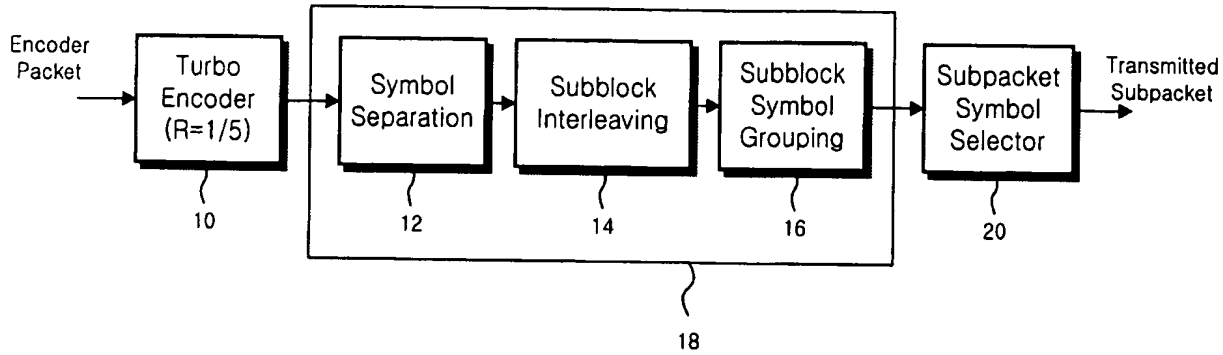
if RAM0_CS=1
    SYS_DATA=RAM0_DATA
else
    SYS_DATA=0
if RAM1_CS=1
    PA0_DATA=RAM2_DATA
else
    PA0_DATA=0
if RAM2_CS=1
    PA1_DATA=RAM1_DATA
else
    PA1_DATA=0

```

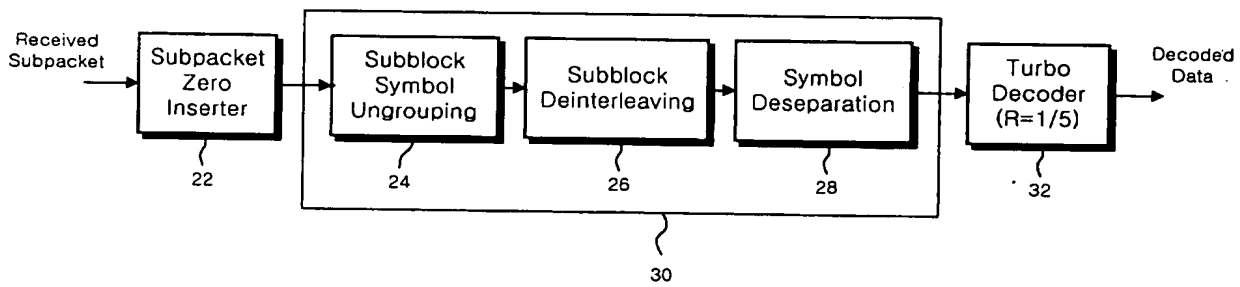
【수학식 12】

【도면】

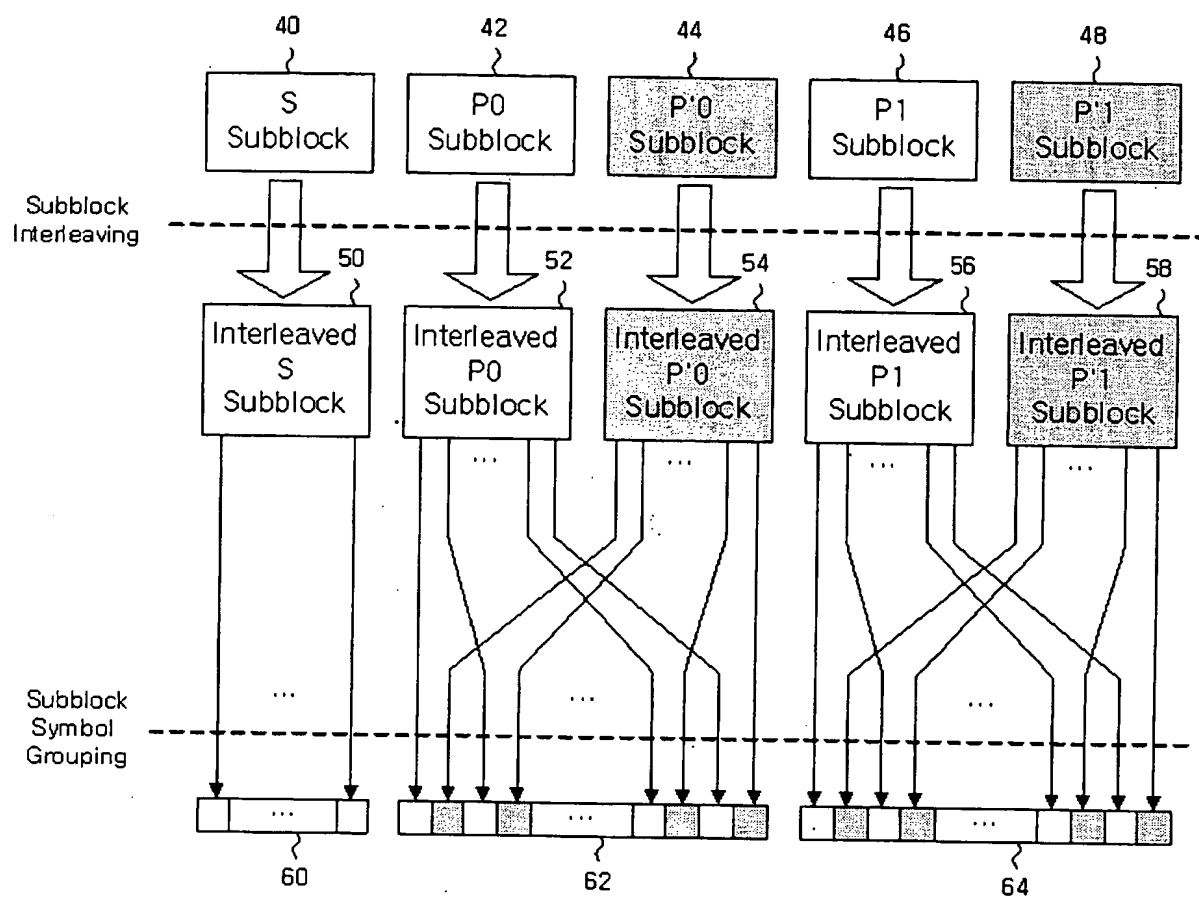
【도 1】



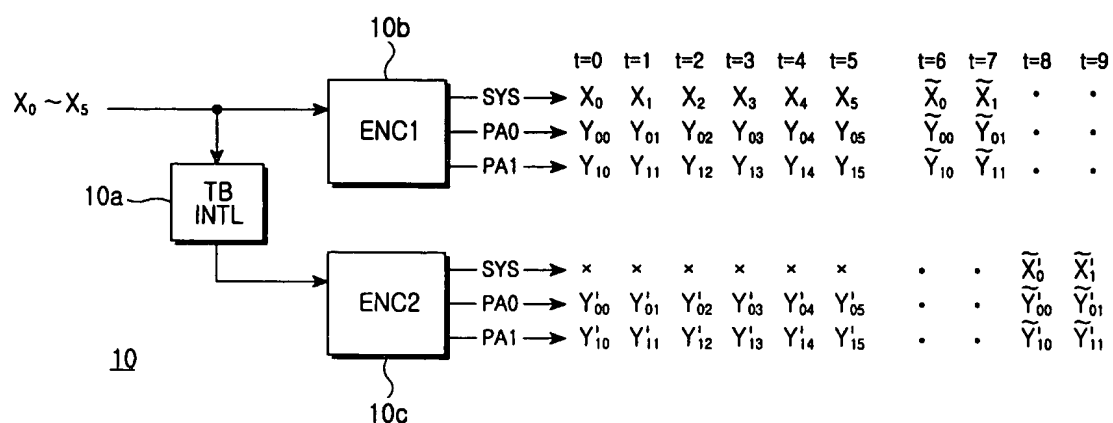
【도 2】



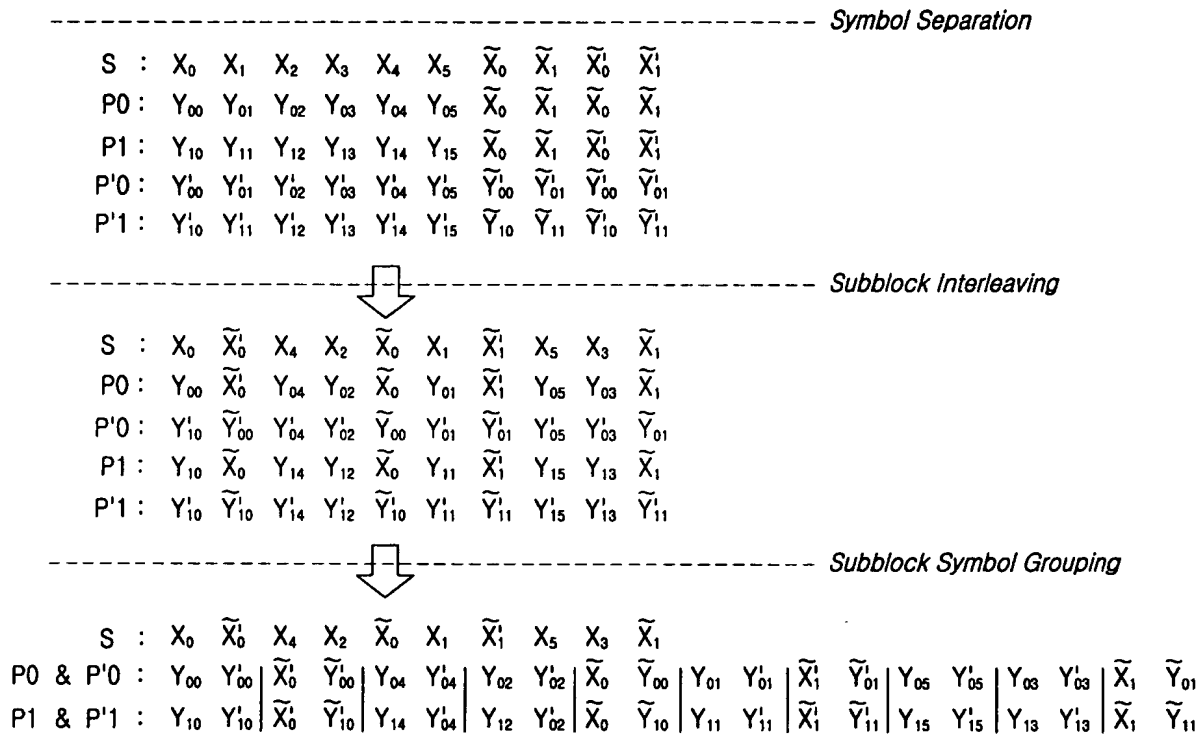
【도 3】



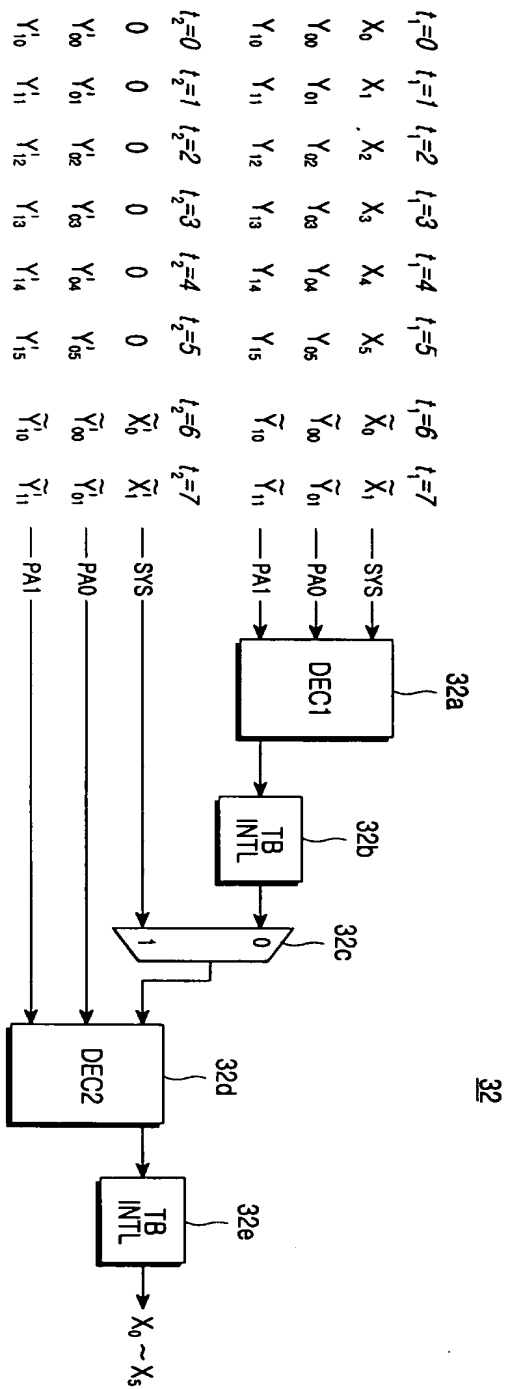
【도 4】



【도 5】



【도 6】



【도 7】

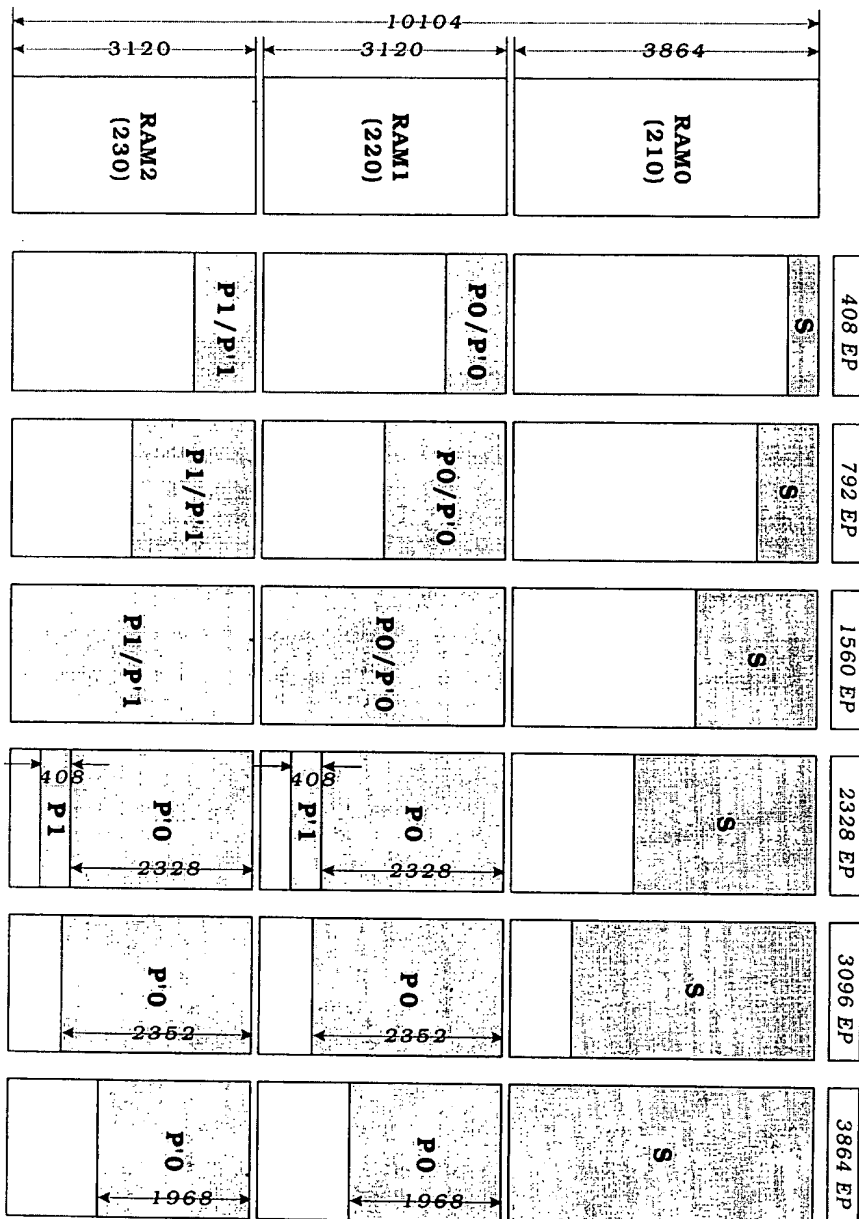
EP Size	Possible Kinds of Code Symbols in Each EP Size	Code Rate
408	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>408</div> <div>816</div> <div>816</div> </div>	0.2000
792	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>792</div> <div>1584</div> <div>1584</div> </div>	0.2000
1560	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>1560</div> <div>3120</div> <div>3120</div> </div>	0.2000
2328	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>2328</div> <div>2328</div> <div>2328</div> <div>408</div> <div>108</div> </div>	0.2985
3096	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>3096</div> <div>2352</div> <div>2352</div> </div>	0.3969
3864	<div> <div>S</div> <div>P0/P'0</div> <div>P1/P'1</div> <div>3864</div> <div>1968</div> <div>1968</div> </div>	0.4954

☐ Systematic

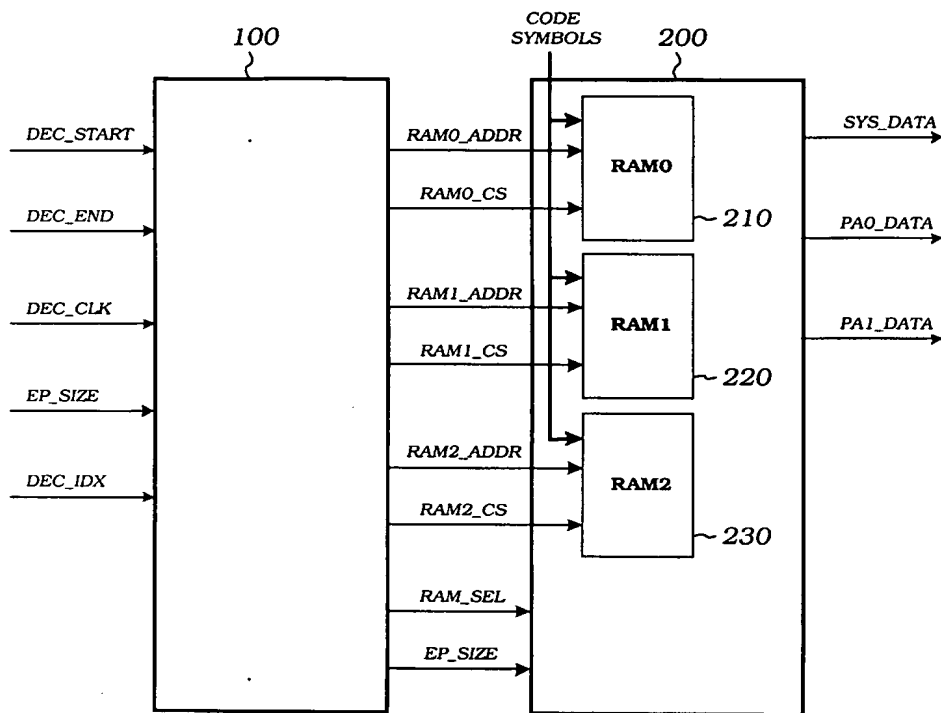
☐ Parity0 / Parity'0

☒ Parity1 / Parity'1

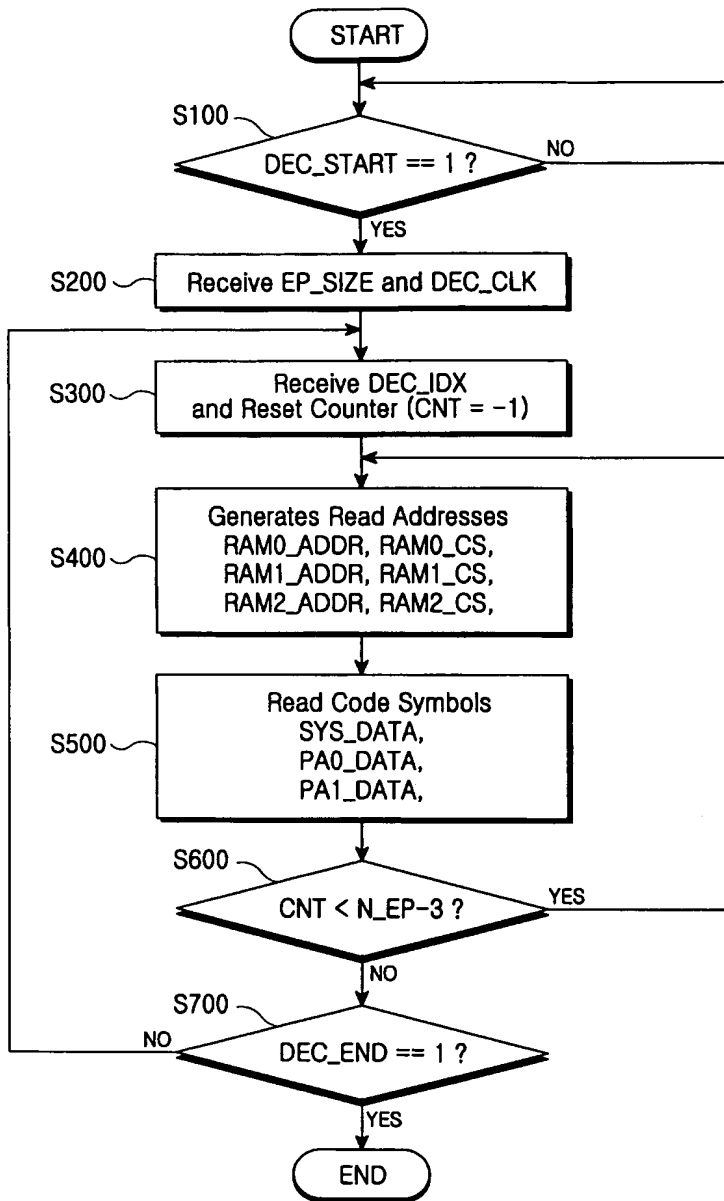
【도 8】



【도 9】

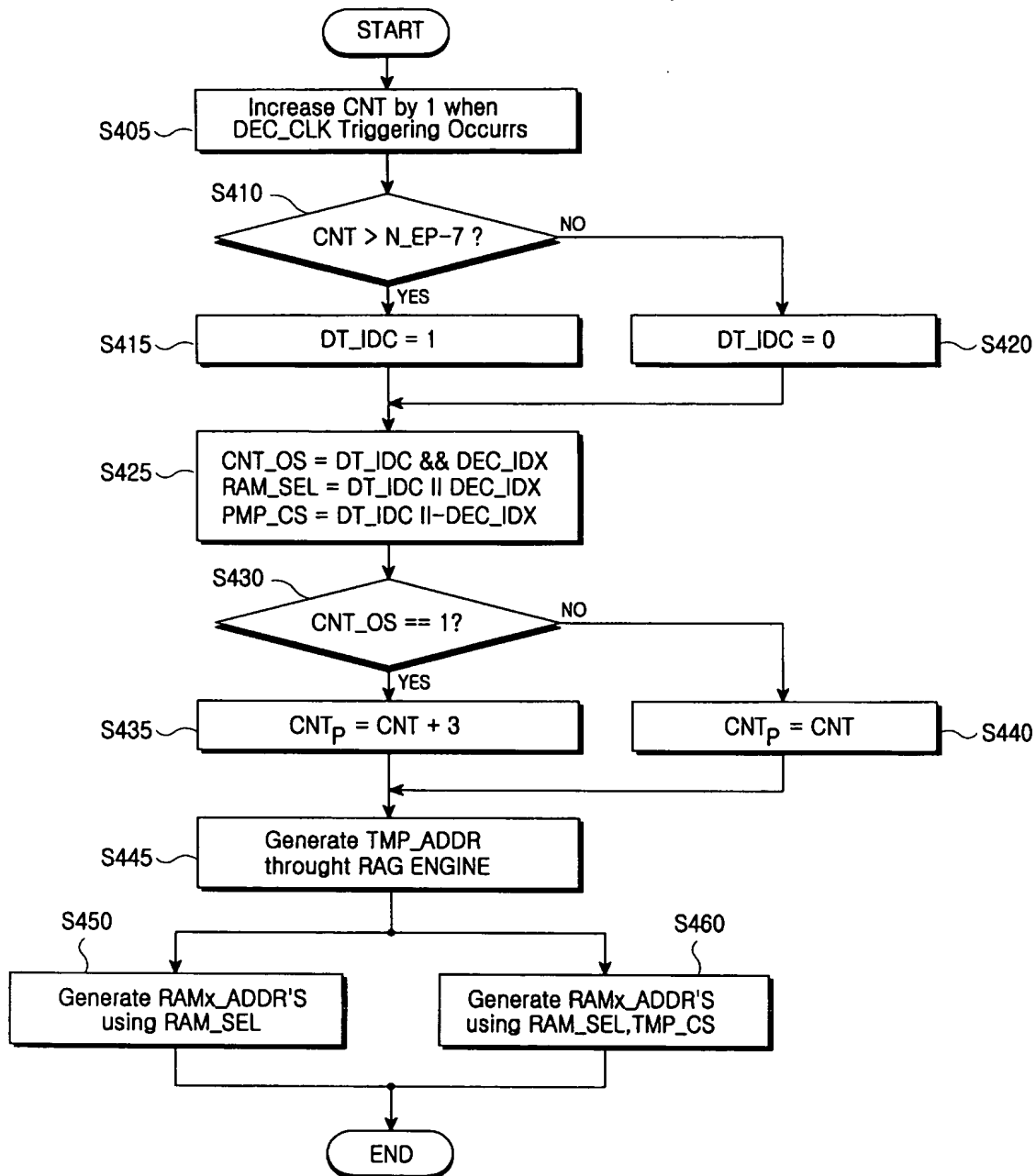


【도 10】

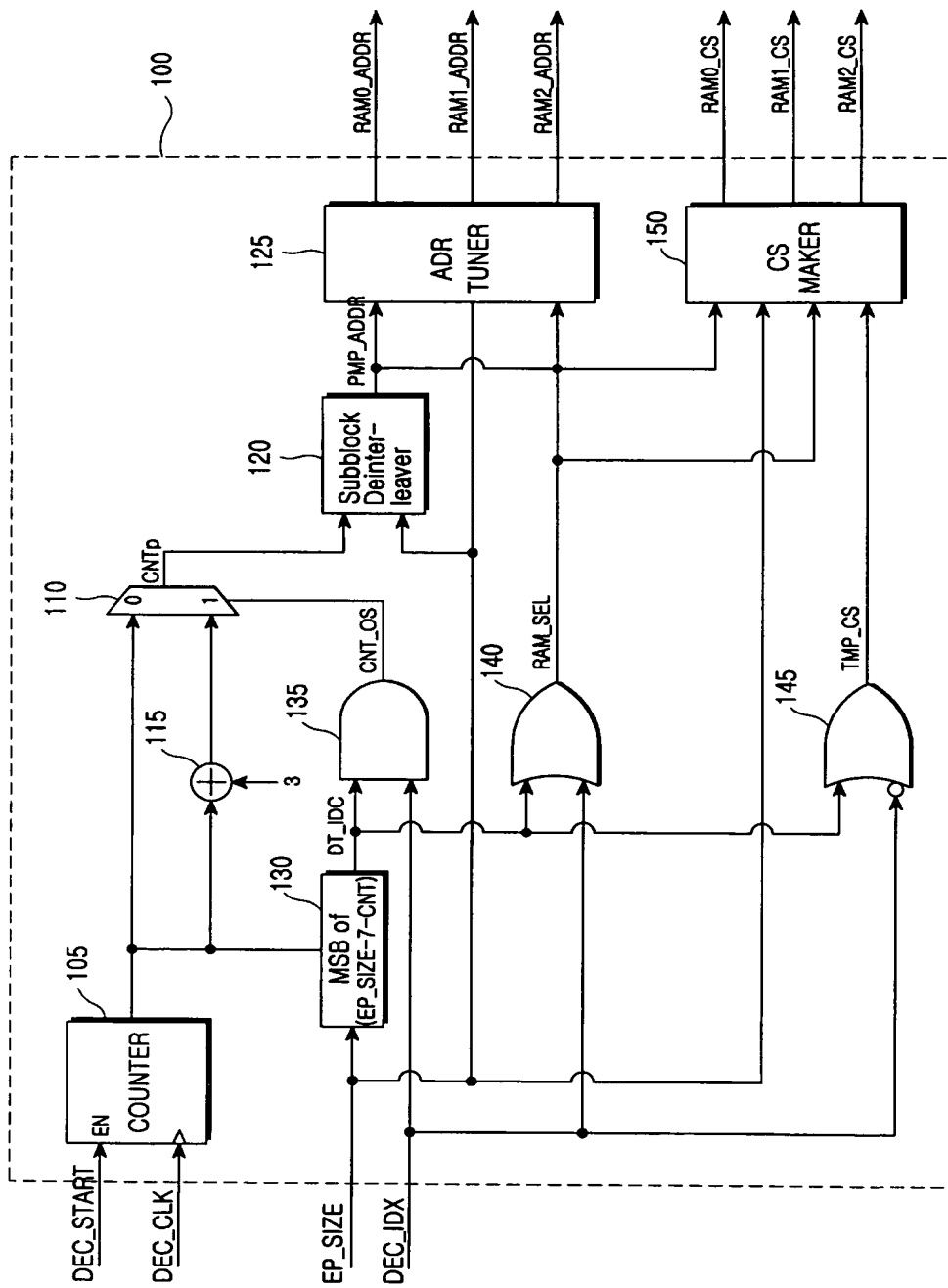


【도 11】

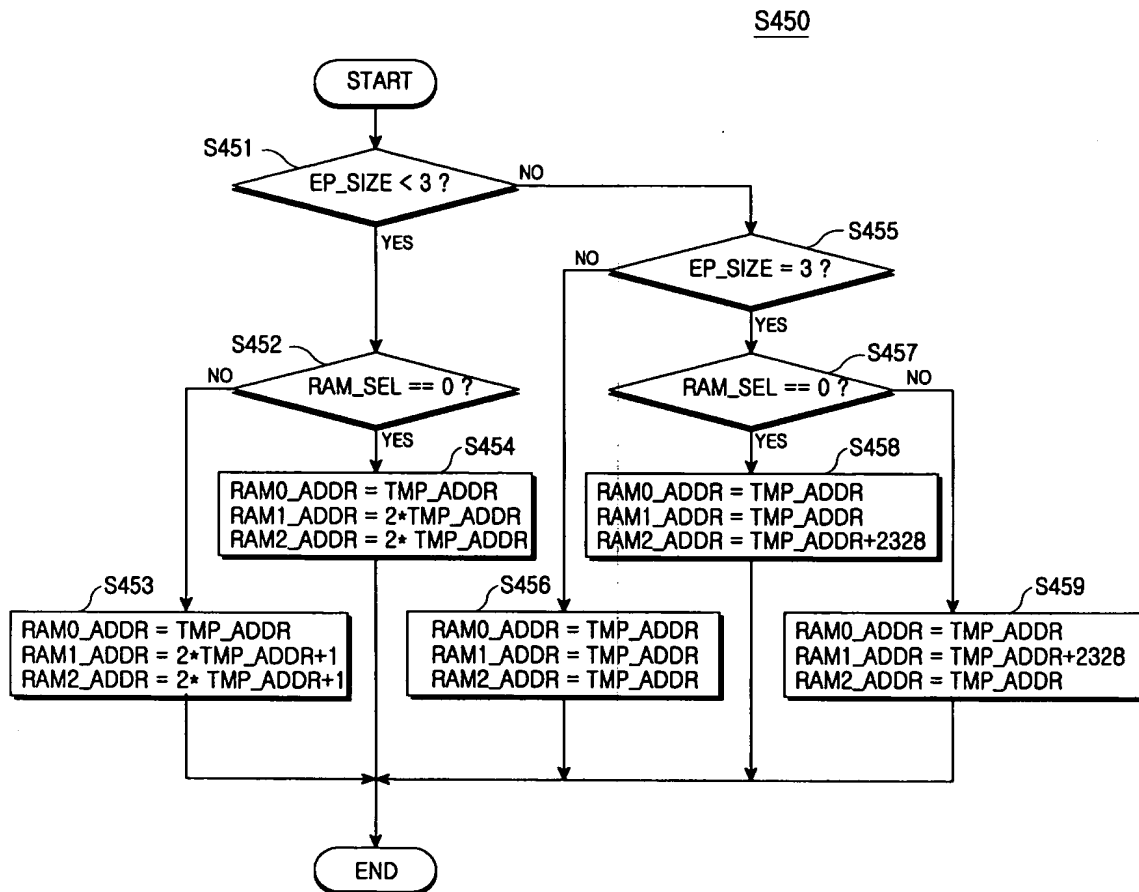
S400



【도 12】

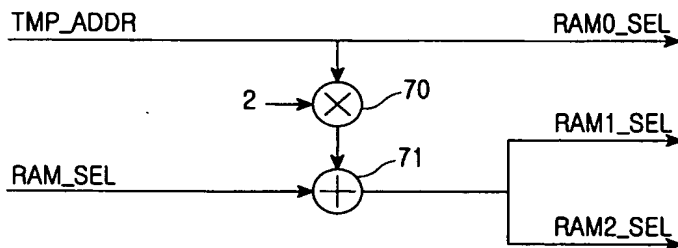


【도 13】



【도 14a】

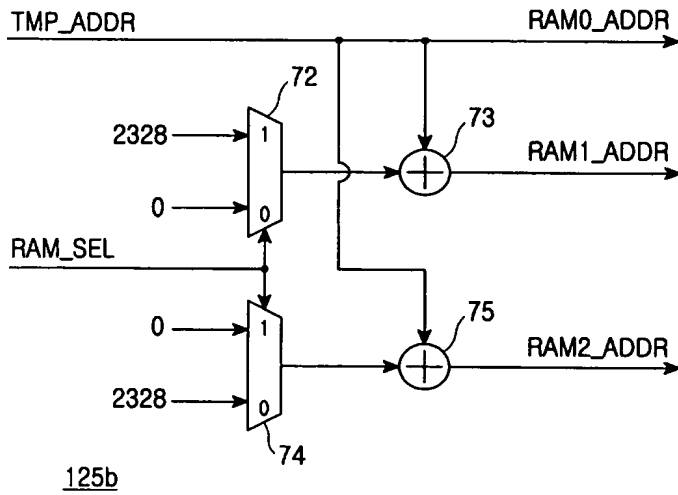
For EP_SIZE(N_EP) = 000(408), 001(792), or 010(1560)



125a

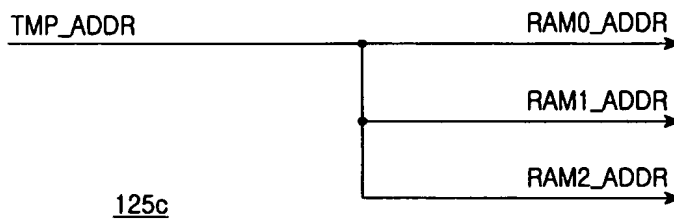
【도 14b】

For EP_SIZE(N_EP) = 011(2328)

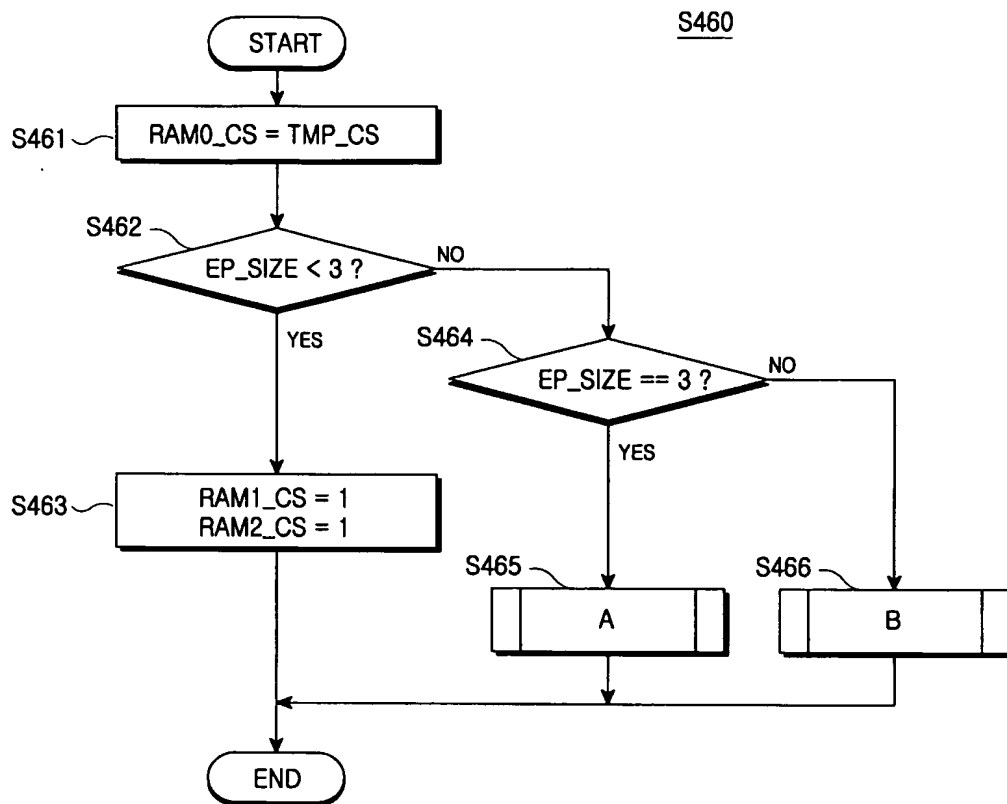


【도 14c】

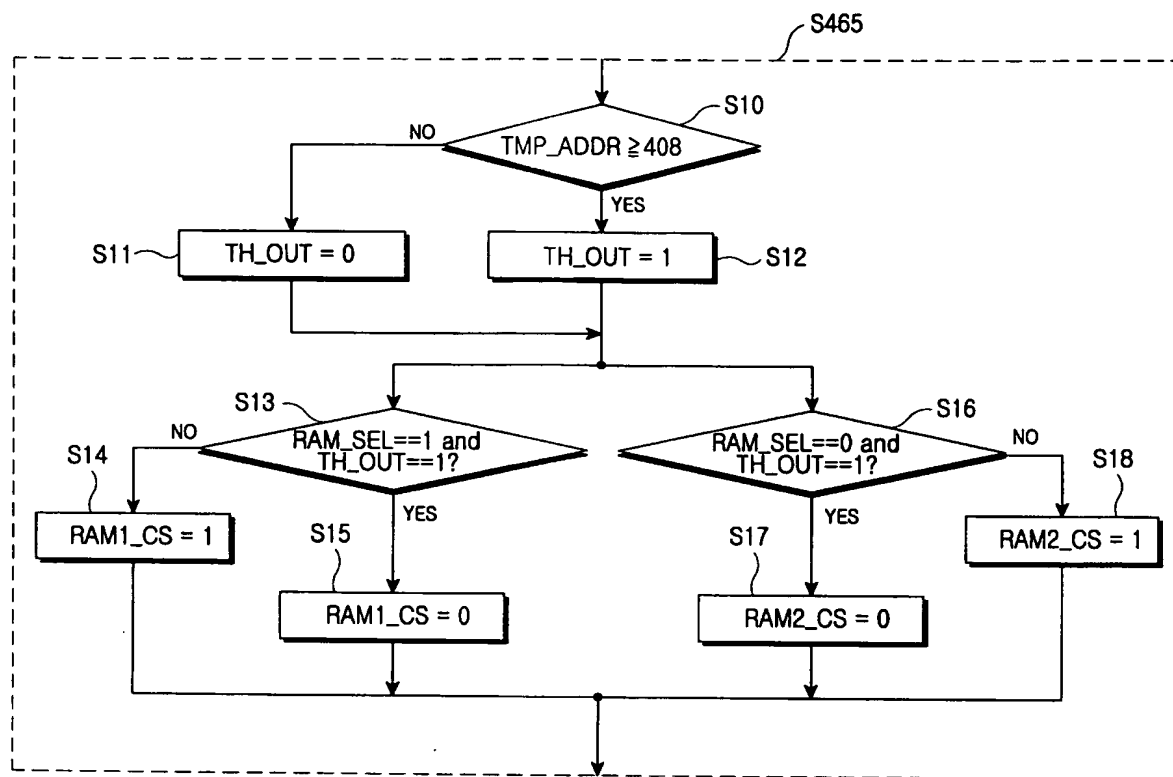
For EP_SIZE(N_EP) = 100(3096), or 101(3864)



【도 15】

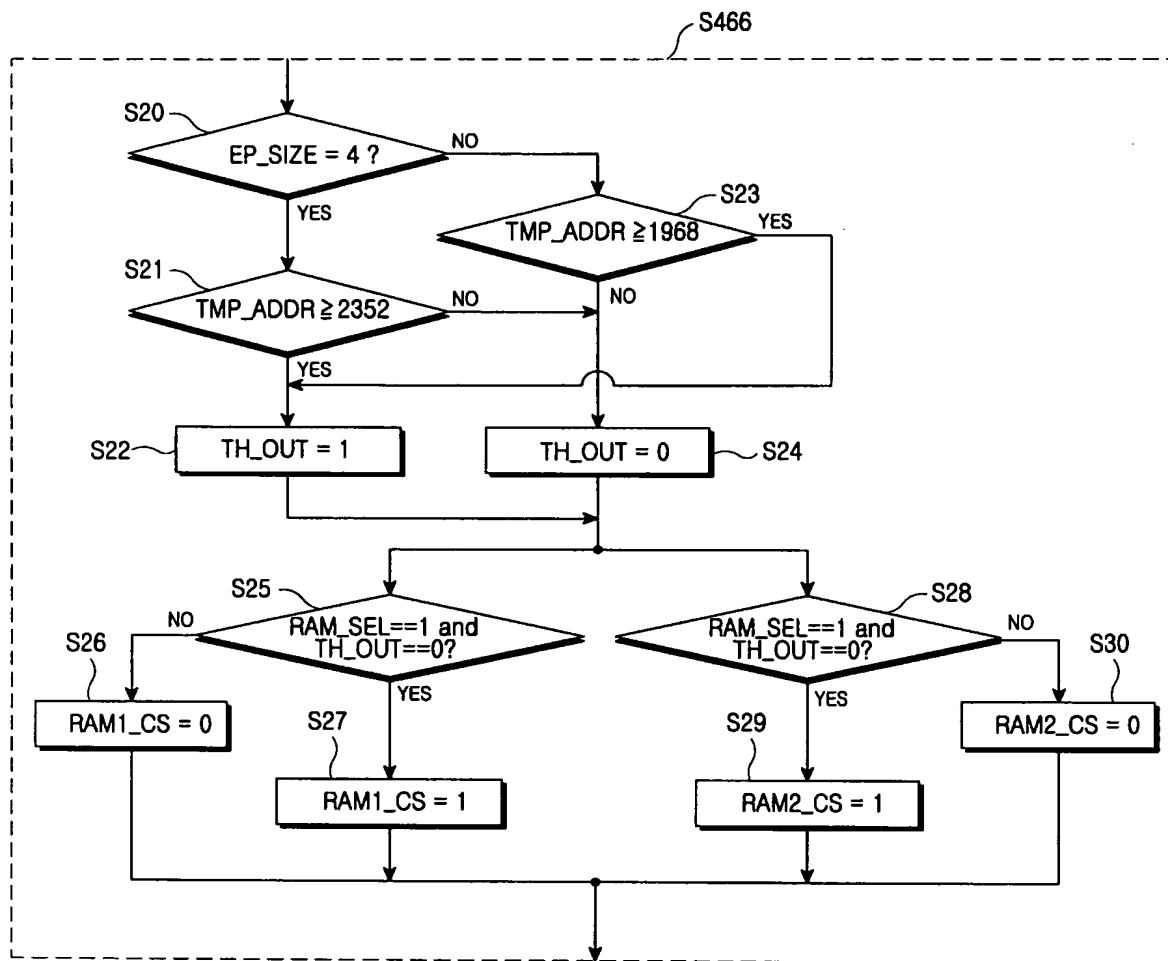


【도 16】



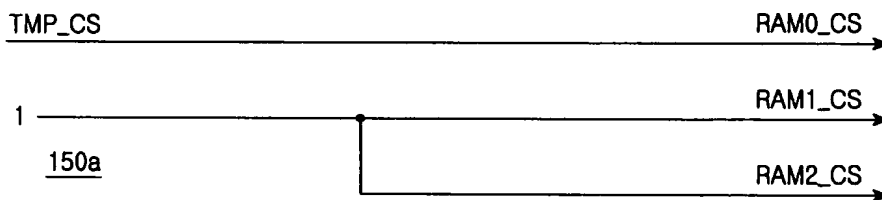


【도 17】



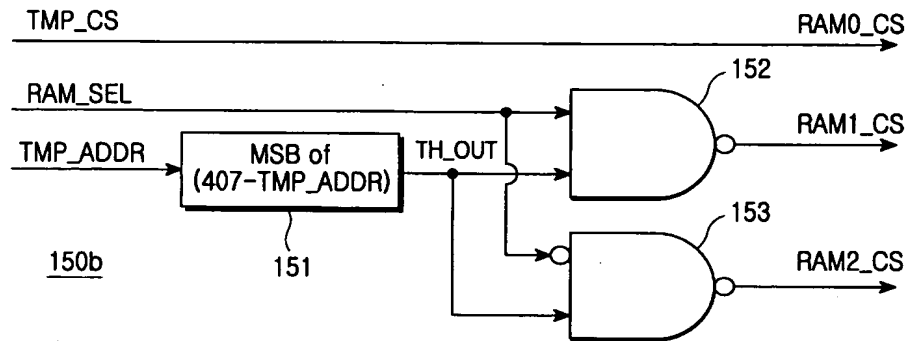
【도 18a】

For EP_SIZE(N_EP) = 000(408), 001(792), or 010(1560)



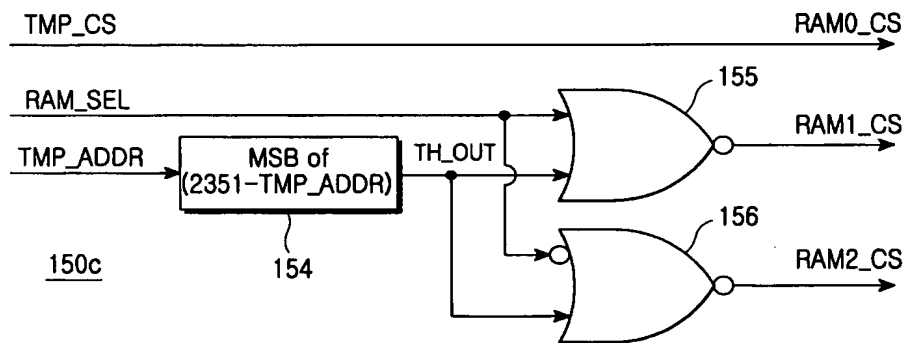
【도 18b】

For EP_SIZE(N_EP) = 011(2328)



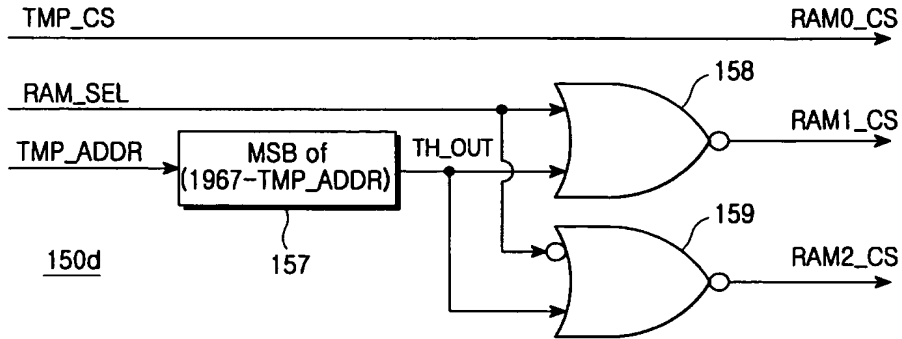
【도 18c】

For EP_SIZE(N_EP) = 100(3096)

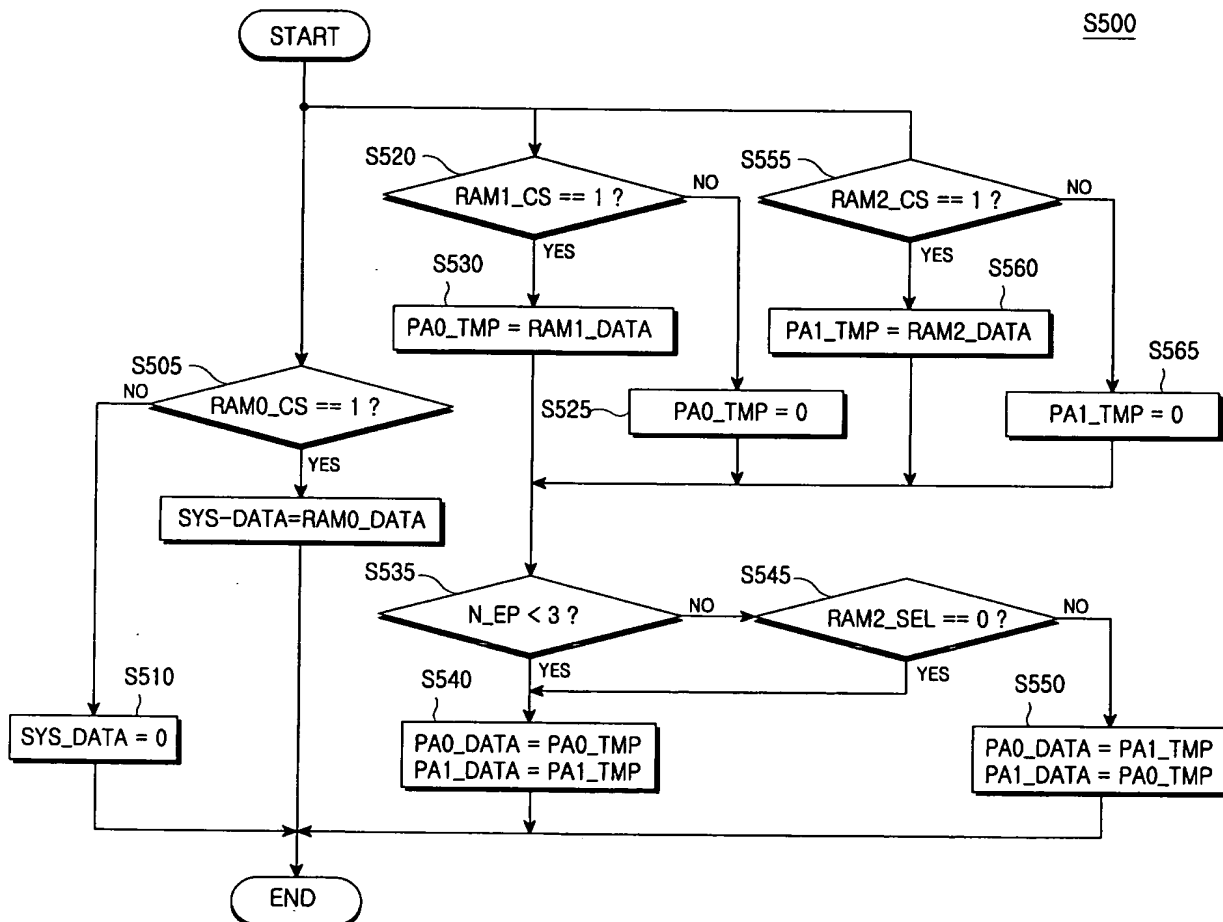


【도 18d】

For EP_SIZE(N_EP) = 101(3864)



【도 19】



【도 20】

